REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE June 1995	3. REPORT TYPE AND DATES COVERED Final 15 May 94 - 14 May 95

4. TITLE AND SUBTITLE

Conference on Computing Science & Statistics
Symposium on the Interface

5. FUNDING NUMBERS

DAAH04-94-G-0222

6. AUTHOR(S)

Edward Wegman (principal investigator)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Interface Foundation of North America, Inc.
Fairfax Station, VA 22039-7460

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 32588.1-MA-CF

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The 26th Symposium on the Interface of Computing Science and Statistics was held on June 15-18, 1994 at the Sheraton Imperial Hotel in Research Triangle Park, NC. The conference theme was *"Computationally Intensive Statistical Methods."* The theme is especially appropriate as computational power has increased dramatically in the last few years and the use of resampling techniques has boomed.

The Interface was scheduled between two other statistics conferences in the same area: the Spring Research Conference on Statistics in Industry, hosted by the National Institute of Statistical Sciences, and the Third World Congress--Bernoulli Society--IMS meetings, at the University of North Carolina in Chapel Hill.

The conference attracted 365 attendees. There were 23 invited sessions, 21 contributed paper sessions, 9 poster presentations, 4 short courses, 2 practical tutorials, several statistical tutorial sessions, one keynote speech, one banquet presentation, and 2 tours.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL

Computing Science and Statistics

Volume 26

Computationally Intensive Statistical Methods

Editors

John Sall

Ann Lehman

Proceedings of the 26th Symposium on the Interface

Research Triangle NC

Interface '94

INTERFACE FOUNDATION OF NORTH AMERICA

PUBLISHER'S FORWARD

Notices

The papers in this volume are printed exactly as they were submitted as a record of the conference and are reproduced as received from the authors. These presentations are presumed to be essentially as given at the 26th Symposium on the Interface. The papers have not been reviewed and no claims are made by the editors or publishers as to the originality or accuracy of their contents.

An example of the recommended citation of articles from this publication is

> Heavlin, W.D. and Finnegan, G.P. (1994), "Dual space algorithms for designing space-filling experiments," ***Computing Science and Statistics***, 26, 41-47.

If more details are required, the editors and the publisher (Interface Foundation of North America, Inc.) may be added.

Purchase of Previous Volumes

You may purchase this Volume and Volumes 20 through 25 (1988 through 1993) from

Interface Foundation of North America, Inc.
P.O. Box 7460
Fairfax Station, VA 22039-7460.

Volume 22 (1990) is also available from

Springer-Verlag, New York, Inc.
175 Fifth Avenue
New York, NY 10010-3402

Volumes 18, 19, 20 and 21 (1986-1989) are available from

American Statistical Association
1429 Duke Street
Alexandria, VA 22314-3402

Interface '95

Please plan to attend the next Interface Symposium scheduled for June 21-24 in Pittsburgh. It will be hosted by Carnegie Mellon University and the Pennsylvania State University with Michael Meyer and James Rosenberger as joint program chairs. For details:

email: interface95@stat.cmu.edu
phone: (412) 268-3108 fax: (412) 268-7828
mail: Interface '95
Department of Statistics
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA

ISBN 1-886658-00-5
PRINTED IN THE UNITED STATES OF AMERICA (1994)

PREFACE

1994 Interface Proceedings

The 26th Symposium on the Interface of Computing Science and Statistics was held on June 15-18, 1994 at the Sheraton Imperial Hotel in Research Triangle Park, NC. The conference theme was *"Computationally Intensive Statistical Methods."* The theme is especially appropriate as computational power has increased dramatically in the last few years and the use of resampling techniques has boomed.

The Interface was scheduled between two other statistics conferences in the same area: the Spring Research Conference on Statistics in Industry, hosted by the National Institute of Statistical Sciences, and the Third World Congress--Bernoulli Society--IMS meetings, at the University of North Carolina in Chapel Hill.

The conference attracted 365 attendees. There were 23 invited sessions, 21 contributed paper sessions, 9 poster presentations, 4 short courses, 2 practical tutorials, several statistical tutorial sessions, one keynote speech, one banquet presentation, and 2 tours.

Conference Events

The conference started Wednesday afternoon with 4 short courses, followed by a mixer that evening. The short courses were organized by Tom Devlin, who is continuing education coordinator for the Statistical Computing Section of ASA. The courses were: *Modern Nonparametric Regression and Classification,* by Trevor Hastie and Rob Tibshirani, *Resampling-Based Multiple Testing,* by P. H. Westfall and S. S. Young, *Algorithms for Estimation and Visualization of Multivariate Density Functions with Applications to Clustering,* by David W. Scott , and *Data Analysis using Interactive Dynamic Graphics: An Introduction to XGobi*, by Di Cook, Martin Koschat, and Deborah Swayne.

On Thursday morning, the keynote address was presented by G. W. "Pete" Stewart professor in the Computer Science Department and Research Professor in the Institute for Advanced Computer Studies at the University of Maryland. Pete talked about "Gauss, Statistics, and Gaussian Elimination," in which Gauss is seen as a statistician inventing numerical methods in the service of fitting data. Pete Stewart is a well-known authority in the field of numerical linear algebra. Originally a student of Alston Householder, he is the author of over ninety papers on various aspects of numerical analysis and matrix computation. His books include *Introduction to Matrix Computation* and, with J. G. Sun, *Matrix Perturbation Theory.* He is a co-author of the LINPACK package for linear algebra. Pete was introduced by Bob Funderlic, North Carolina State University.

On Thursday evening, there were tours to the UNC Graphics and Image Lab in Chapel Hill, and to SAS Institute in Cary. The feature at the UNC lab was virtual reality and the Pixelplanes 5 parallel graphics computer. The feature at SAS Institute was the new 400,000 square foot research building.

On Friday, a banquet dinner was held with music by the Bluegrass Retreat. Interface business manager Ruth Lee played bass guitar. Dinner was followed by a presentation on computer animation by Wayne Lytle, an award-winning computer graphics animator from the Cornell University Theory Center. Wayne's presentation featured scientific animations describing the recent breakthrough discovery of planets in a distant star system. Particularly enjoyable were a humorous animation on glitziness overload in scientific presentations, and a segment on music animation.

The Conference Organization

Interface Conferences are sponsored by the Interface Foundation of North America. IFNA is a nonprofit educational corporation founded in 1987 to sponsor the symposium and publish the proceedings. IFNA also co-publishes the *Journal of Computational and Graphical Statistics.*

The conference is undertaken with the support and cooperation of the following societies: the American Statistical Association, the Institute for Mathematical Statistics, the International Association for Statistical Computing, the Society for Industrial and Applied Mathematics, and the Operations Research Society of America.

SAS Institute hosted this year's conference, with John Sall serving as program chair. SAS Institute is a software company specializing in statistical computing, and is located in nearby Cary, NC. SAS Institute provided personnel and services free of charge for the meeting.

The program committee and session organizers were Stephen G. Eick, J. S. Marron, Russ Wolfinger, Sally Morton, Mike West, S. Stanley Young, Raoul LePage, Ron Gallant, Alex Georgiev, Bill DuMouchel, Cyrus R. Mehta, Chris Portier, Ed Wegman, David Rocke, Iain Johnstone, Peter Munson, Tim Hesterberg, Richard Smith, Francoise Seillier-Moiseiwitsch, Forrest Young, and John Elder. Featured speakers included Adrian Smith, Andrew Barron, and Mary Ellen Bock. Additional tutorials were given by Tim Arnold and Phil Spector.

Session chairs included Jianqing Fan, Lisa LaVange, James L. Rosenberger, Mark Little, Nick Fisher, Ming Tan, Wolfgang Hartmann, John Elder, Dave Dickey, Karen Kafadar, Phil Spector, Warren Sarle, John Nash, George Guirguis, Al Best, Forrest Young, Alan Genz, Phil Spector, Mary Ellen Bock, Cyrus R. Mehta, Chris Portier, Leonard B. Hearne, Bill Kemple, Feng Gao, Ying So, Deborah Swaine, Dennis Boos, and Gordon Johnston.

Outside of the program, the people that put the conference together were: Ruth Lee, conference business manager, Susan Byrd, hotel coordinator, Armistead Sapp, equipment manager, Jane Pierce, abstracts editor, Stefanie Barber Mueller, Kristin Rinne, Marybeth Mahoney, Curt Yeo and SAS Institute Copy Center, for graphic arts, Lynn Fountain, Chris Gilmore, Bob Rodriguez for the SAS tour, Linda Houseman for the UNC tour. Interpath provided Internet connections. The IFNA head office with Ed Wegman and Pat Joyce did the printing, mailing, grant administration, and accounts payable.

John Sall and Ann Lehman
Editors

Please plan to attend the next Interface Conference, scheduled for June 21-24 in Pittsburgh. It will be hosted by Carnegie Mellon University and the Pennsylvania State University with Michael Meyer and James Rosenberger as joint program chairs. For details:

email: interface95@stat.cmu.edu
Phone: (412) 268-3108 Fax: (412) 268-7828
Mail: Interface '95
Department of Statistics
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA.

HOST

SAS Institute Inc.
Cary, North Carolina

SPONSER OF THE 26th SYMPOSIUM ON THE INTERFACE

Interface Foundation of North America

COOPERATING SOCIETIES AND INSTITUTIONS

American Statistical Association (ASA)
Institute of Mathematical Statistics (IMS)
International Association for Statistical Computing (IASC)
Society for Industrial and Applied Mathematics (SIAM)
Operations Research Society of America (ORSA)

EXHIBITORS

SLP Statistiques Logiciels Pedagogie
Trilogy Consulting Corporation
Chapman and Hall
Elsevier Science
Springer–Verlag
StaSci–MathSoft, Inc.
The Mathworks, Inc.
Society for Industial and Applied Mathematics
Interface Foundation of America
John Wiley & Sons, Inc.
BBN Inc.

TABLE OF CONTENTS

Keynote Address

Fast Implementations of Smoothers

Longitudinal and Mixed Models

Contributed Papers 1: Experimental Design

Contributed Papers 2: Fractal, Neural, Other

Poster Sessions

Green Thumbs: Extensions and Applications of Tree Modeling Methods

Bayesian Curve Fitting

Space Filling Experimental Designs: Theory, Computer Construction and Analysis

Contributed Papers 3: Longitudinal

Contributed Papers 4: Computing

Contributed Papers 5: Enhancements to Tree Algorithms

Contributed Papers 6: Multiple Comparisons

Contributed Papers 7: Smoothers and Nonparametric Regression

Contributed Papers 8: Density Estimation

Contributed Papers 9: Numerical

Contributed Papers 10: Trees II

Nonparametric Regression for Edge and Peak Preserving

Software for MetaAnalysis

Special Contributed Papers 11: Visual Statistical Analysis

Contributed Papers 12: Gibbs Samplers

Contributed Papers 13: Computing

Smart Monte Carlo Methods for Conditional Infrerence in Exponential Families

Contributed Papers 14: Multivariate

Contributed Papers 15: Software

Robust Regression and Multivariate Analysis

Contributed Papers 16: Genetics

Contributed Papers 17: Bootstrap

Contributed Papers 18: Graphics

Contributed Papers 19: Nonparametrics

Efficient Bootstrap Computations

Convergence of Markov Chain Samplers

Computations Techniques in Genetics and Molecular Biology

Contributed Papers 20: Robust

Contributed Papers 21: Parametric Modeling

Gauss, Statistics, and Gaussian Elimination

G. W. Stewart
Department of Comuter Science and
Institute for Advanced Computer Studies
University of Maryland at College Park

1. Introduction

Everyone knows that Gauss invented Gaussian elimination, and, excepting a quibble, everyone is right.[1] What is less well known is that Gauss introduced the procedure as a mathematical tool to get at the precision of least squares estimates. In fact the computational component in the original description is so little visible, that it takes some doing to see an algorithm in it.

Gaussian elimination, therefore, was not conceived as a general numerical algorithm with applications in statistics and least squares. Rather it was a procedure that sprang from the interface of statistics and computation. Since the full story is known only to the few who have consulted the original sources, I hope my readers will be interested to see how Gauss did things. But there is more than the satisfaction of idle curiosity here. Gauss and Laplace were the premier statisticians of their day, and Gauss alone was the premier numerical analyst. Today we still have something to learn from observing Gauss's practices.

2. Chronicles

The principle of least squares arose from the problem of combining sets of overdetermined equations to form a square system that could be solved for the unknowns. The problem went under the name of the combination of observations, and has been well surveyed by Stigler [23] in his *History of Statistics.* By way of background, I will relate in chronological order the major events in the story of least squares, from Gauss's first discovery to his final treatment in the 1820's.

In his correspondence, Gauss asserted that he had discovered the principle of least squares in 1824 (or 1825, the dates vary). Gauss seems to have had little regard for the principle itself, and even said he thought others must have used it before him. In June of 1828 Gauss [11, v. 10] made the following entry in the little diary of discoveries he kept from 1796 to 1814: "Probability calculus defended against Laplace."[2] Laplace, following Boscovich [1, 16], had suggested that observations be combined by minimizing the sum of the absolute values of the residuals subject to the condition that the residuals sum to zero. Gauss felt that this way of combining observations violated the dictates of probability theory, and his alternative was the first probabilistic justification of least squares.

The following entry in the diary, also dated June 1898, contains the statement: "The problem of elimination resolved in such a way that nothing more can be desired."[3] I take this entry to be the first reference to Gaussian elimination. But a decade was to pass before Gauss published either the probabilistic justification or the elimination procedure.

Although we tend to regard Gauss chiefly as a mathematician, it was as an astronomer that he first made his mark. On New Year's Day of 1801, the astronomer Piazzi discovered the asteroid Ceres. The new planet became unobservable after only nine degrees of an arc had been recorded, and astronomers were faced with problem of determining where to look for it next. Gauss undertook the calculation, using new techniques in physical astronomy and presumably his principle of least squares. At the end of 1801, he predicted where in the heavens the asteroid would be found, and his reputation was made.

Gauss, who was generally slow to publish, began work in 1805 on his *Theoria Motus Corporum Coelestium,* in which he described his techniques for computing orbits and gave his first probabilistic justification of the principle of least squares. He finished in 1806, but his publisher, worried by German losses to Napolean, insisted he translate the treatise into Latin. In consequence it did not appear until 1809 [2]. In the meantime, Legendre [20] published and named the method of least squares (*la méthode des moindres quarrés*) in an appendix to a memoir appearing in 1805. When the *Theoria Motus* finally appeared, Legendre found that Gauss had claimed the principle for his own, and he took exception. The result was a priority dispute, which need not concern us here. [4]

[1] The quibble is that in 1759, in the very first paper to appear in his collected works [14], Lagrange gave the basic computational formulas for Gaussian elimination. His purpose, however, was to determine if a critical point was a minimum, not to solve linear equations. There is no indication that the paper had any influence on Gauss, or anyone else.

[2] In the original Latin: *Calculus probabilitatis contra La Place defensus.*

[3] *Problema eliminationis ita solutum, ut nihil amplius desiderari possit.*

[4] Placket [21] gives balanced survey with translations from

In the *Theoria Motus*, Gauss had assumed the errors in the observations were normally distributed. In 1811, Laplace [17] his central limit theorem give an essentially different justification of least squares. This is not the place to enter into details, but briefly Laplace showed that the solutions of a combination of equations were asymptotically normal and from this concluded that the least squares combination would minimize the mean absolute error in the solutions. Laplace's approach does not readily extend beyond two unknowns.

The final chapter occurred in the 1820's when Gauss [5, 6, 8] published two memoirs on least squares. The first, in two parts, contains yet another justification of least squares — Gauss's famous minimum variance theorem. These papers also contain some nice algorithmics, which will concern us later.

3. The Precision of Estimates

The first appearance of Gaussian elimination in print occurs in Section 182 of the *Theoria Motus.* In order to understand what Gauss is about, we will have to sketch some background.

Gauss (after a linearization) considers the model[5]

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \mathbf{e},$$

where $\mathbf{X}$ is $n \times p$. The errors e_i are assumed to be independent randome variables with common distribution $\varphi(e)$. Gauss introductes the function

$$\varphi(y_1 - \mathbf{x}_1^{\mathrm{T}}\mathbf{b})\varphi(y_2 - \mathbf{x}_2^{\mathrm{T}}\mathbf{b}) \cdots \varphi(y_n - \mathbf{x}_n^{\mathrm{T}}\mathbf{b}), \qquad (3.1)$$

where the $\mathbf{x}_i^{\mathrm{T}}$ are the rows of $\mathbf{X}$ and uses a Bayesian argument with a uniform prior to argue that the value of $\mathbf{b}$ that maximizes (3.1) is the most probable value of the unknowns.

Gauss now supposes the distribution of the e_i is normal; that is, $\varphi(e) \propto e^{-h^2e^2}$. He identifies the parameter h with the precision[6] of $\mathbf{y}$. The function (3.1) now becomes proportional to

$$e^{-h^2\Omega}, \qquad (3.2)$$

where

$$\Omega = (\mathbf{y} - \mathbf{X}\mathbf{b})^{\mathrm{T}}(\mathbf{y} - \mathbf{X}\mathbf{b})$$

is the residual sum of squares. Thus, Gauss's most probable value is obtained by minimizing the residual sum of squares, which justifies the principle of least squares. The normal equations can be derived as usual by differentiation.

Gauss next turns to the problem of estimating the precision of the least squares estimates. His technique is to integrate all but the last unknown out of (3.2), after which the precision can be read off. However, to perform the required integrations Ω must be expressed in a special form, and the tool for arriving at that form is Gaussian elimination.

The procedure as given by Gauss is the following. Let

$$u_1 = \frac{1}{2}\frac{\partial\Omega}{\partial b_1} \equiv r_{11}b_1 + r_{12}b_2 + \cdots + r_{1p}b_p - s_1,$$

and let

$$\Omega_1 = \Omega - \frac{u_1^2}{r_{11}}.$$

Then clearly the derivative of Ω_1 with respect to b_1 is zero, so that Ω_1 is independent of b_1.

One more step will illustrate the general procedure. Set

$$u_2 = \frac{1}{2}\frac{d\Omega_1}{db_2} \equiv r_{22}b_2 + r_{23}b_3 + \cdots + r_{2p}b_p - s_2.$$

Then

$$\Omega_2 = \Omega_1 - \frac{u_2^2}{r_{22}}$$

is independent of b_1 and b_2. Continuing in this manner we arrive at the decomposition

$$\Omega = \frac{u_1^2}{r_{11}} + \frac{u_2^2}{r_{22}} + \cdots + \frac{u_p^2}{r_{pp}} + \rho,$$

in which u_i is independent of $b_1, \ldots, b_{i-1}$ and ρ is constant.

Gauss now considers the expression

$$e^{-h^2\Omega} = \exp\left(-h^2\frac{u_1^2}{r_{11}}\right) \cdot \exp\left(-h^2\frac{u_2^2}{r_{22}}\right) \cdots \exp\left(-h^2\frac{u_p^2}{r_{pp}}\right).$$

and integrates with respect to b_1 over the real line. Since the last $p-1$ factors in this expression are free of b_1, they remain unchanged by the integration. The first factor integrates to a constant. Thus Gauss is left with a distribution proportional to

$$e^{-h^2\Omega_1} = \exp\left(-h^2\frac{u_2^2}{r_{22}}\right) \cdots \exp\left(-h^2\frac{u_p^2}{r_{pp}}\right),$$

which is free of b_1. Continuing this process of integrating out the parameters b_i, Gauss finds that the distribution of b_p is proportional to

$$\exp\left(-h^2\frac{u_p^2}{r_{pp}}\right),$$

Gauss's correspondence.

[5] We will make free use of matrices in what follows, but only as means of abbreviating Gauss's scalar equations.

[6] We must not use terms like variance or standard deviation here. The number h is simply a parameter in a specific distribution. Only in the *Theoria Combinationis* will Gauss introduce the second moment of a general distribution as a measure of variation

where

$$u_p = r_{pp}b_p - s_p.$$

Gauss concludes that the most probable value of b_p, obtained by setting $u_p = 0$, is

$$\hat{b}_p = \frac{s_p}{r_{pp}}$$

and its precision is

$$\frac{h}{\sqrt{r_{pp}}}.$$

Gauss now goes on to show that if you write the normal equations in the form

$$\mathbf{Ab} = \mathbf{c} \tag{3.3}$$

and express $\mathbf{b}$ as a function of $\mathbf{c}$ in the form

$$\mathbf{b} = \mathbf{Vc}, \tag{3.4}$$

then the (p,p)-element of $\mathbf{V}$ is $\frac{1}{r_{pp}}$. Since the resulting expression for the precision clearly does not depend on the position of the unknown, Gauss concludes that the precision of *any* of the estimates $\hat{b}_i$ is $h\sqrt{v_{ii}}$.

It is ironic that the *Theoria Motus* should have become the principle reference for Gaussian elimination as a computational tool. As we have seen, Gauss used elimination to give a derivation of one of the most important results of linear regression theory. He was certainly aware of the computational consequences of his elimination procedure, and promises to describe them in a later work. But computational considerations are absent from the *Theoria Motus* itself. Gauss merely points out that the normal equations can be solved by ordinary elimination (*eliminatio vulgaris*), presumably a variant of what we now call Gauss–Jordan elimination. An extension, which Gauss will later call general elimination (*eliminatio indefinite*), can be used to pass from the normal equations (3.3) to the inverse system (3.4).

4. The Scalar Connection

In 1810, in *Disquisitio de Elementis Ellipticis Palladis* [3], Gauss gave the numerical details of his algorithm and illustrated it with an example. The formulas can be derived by observing that a homogeneous quadratic form is determined by its matrix of second derivatives. Specifically, if we set

$$a_{ij} = \frac{1}{2}\frac{\partial^2\Omega}{\partial b_i \partial b_j},$$

then it follows from the formula

$$\Omega_1 = \Omega - \frac{1}{2a_{11}}\frac{\partial\Omega}{\partial b_1}$$

that

$$a_{ij}^{(1)} \equiv \frac{1}{2}\frac{\partial^2\Omega_1}{\partial b_i \partial b_j} = a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}}.$$

In the expression on the right, we recognize the formulas for performing one step of Gaussian elimination, as we understand it today, on a matrix whose elements are a_{ij}. This is essentially the algorithm Gauss describes in the *Disquisitio.*

To complete the solution of the normal equations by Gaussian elimination, note that since

$$\Omega = \frac{u_1^2}{r_{11}} + \frac{u_2^2}{r_{22}} + \cdots + \frac{u_p^2}{r_{pp}} + \rho,$$

the function Ω assumes its minimum value ρ when

$$u_1 = u_2 = \cdots = u_p = 0.$$

Since

$$0 = u_p = r_{pp}b_p - s_p$$

is a linear equation involving only b_p, it can be solved immediately for b_p. Having determined b_p, one can solve for b_{p-1} from the equation

$$0 = u_{p-1} = r_{p-1,p-1}b_{p-1} + r_{p-1,p}b_p - s_{p-1}.$$

Continuing in this manner, we can determine estimates for all the unknown b_p. This of course is nothing more than the back substitution phase of Gaussian elimination.

5. The Matrix Connection

The above description of the algorithm is incomplete, in the sense that it does not give formulas for the constant parts s_i of the functions u_i. To see where they come from, it will be useful to express the algorithm in terms of matrices.

The function Ω can be written in the form

$$\begin{aligned}\Omega &= (\mathbf{b}^{\mathrm{T}}\ \ -1)\begin{pmatrix}\mathbf{X}^{\mathrm{T}}\mathbf{X} & \mathbf{X}^{\mathrm{T}}\mathbf{y}\\ \mathbf{y}^{\mathrm{T}}\mathbf{X} & \mathbf{y}^{\mathrm{T}}\mathbf{y}\end{pmatrix}\begin{pmatrix}\mathbf{b}\\ -1\end{pmatrix}\\ &\equiv (\mathbf{b}^{\mathrm{T}}\ \ -1)\begin{pmatrix}\mathbf{A} & \mathbf{c}\\ \mathbf{c}^{\mathrm{T}} & \eta\end{pmatrix}\begin{pmatrix}\mathbf{b}\\ -1\end{pmatrix}\end{aligned}$$

If we set

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1p}\\ 0 & r_{22} & \cdots & r_{2p}\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \cdots & r_{pp}\end{pmatrix} \quad\text{and}\quad \mathbf{s} = \begin{pmatrix}s_1\\ s_2\\ \vdots\\ s_p\end{pmatrix},$$

then it is easy to verify that

$$\begin{pmatrix}\mathbf{A} & \mathbf{c}\\ \mathbf{c}^{\mathrm{T}} & \eta\end{pmatrix} = \begin{pmatrix}\mathbf{R}^{\mathrm{T}} & 0\\ \mathbf{s}^{\mathrm{T}} & \rho\end{pmatrix}\begin{pmatrix}\mathbf{D}^{-1} & 0\\ 0 & \rho^{-1}\end{pmatrix}\begin{pmatrix}\mathbf{R} & \mathbf{s}\\ 0 & \rho\end{pmatrix},$$

where

$$\mathbf{D} = \operatorname{diag}(r_{11}, r_{22}, \ldots, r_{pp}).$$

Thus Gaussian elimination, as practiced by Gauss, amounts to factoring the *augmented cross-product matrix* into a lower triangular matrix, a diagonal matrix, and the transpose of the lower triangular matrix. It is common practice today to work with the augmented cross-product matrix.

The vector **u** whose components are the functions u_i can be written in the form

$$\mathbf{u} = \mathbf{R}\mathbf{b} - \mathbf{s}.$$

The process sketched above of setting the u_i to zero and back-solving amounts to solving the triangular system

$$\mathbf{R}\mathbf{b} = \mathbf{s}.$$

6. The Computation of Variances

Writing in 1821, Gauss [4] summarized his and Laplace's justifications of least squares as follows.

> From the foregoing we see that the two justifications each leave something to be desired. The first depends entirely on the hypothetical form of the probability of the error; as soon as that form is rejected, the values of the unknowns produced by the method of least squares are no more the most probable values than is the arithmetic mean in the simplest case mentioned above. The second justification leaves us entirely in the dark about what to do when the number of observations is not large. In this case the method of least squares no longer has the status of a law ordained by the probability calculus and has only the simplicity of the operations it entails to recommend it.

In the *Pars Prior* of his memoir *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae* [7], Gauss resolved the dilemma by introducing the notion of mean square error as a measure of variance and showing that among all linear combinations of the observations that produced exact estimates in the absence of error the least squares estimates have least mean square error.

In the *Pars Posterior* of the *Theoria Combinationis* [6], Gauss addresses the problem of computing variances. He points out that his elimination method gives only the variance of the last unknown. Since (he continues) a general elimination to invert the normal equations is expensive, some calculators have adopted the practice of performing the elimination with another unknown placed last.[7] Gauss says that he will give a better way.

Gauss actually gives two solutions to the problem. In the first he shows that if one inverts the system $\mathbf{R}\mathbf{b} = \mathbf{s}$ to get $\mathbf{T}\mathbf{s} = \mathbf{b}$, then the matrix **V** obtained by passing from (3.3) to (3.4) can be written

$$\mathbf{V} = \mathbf{T}\mathbf{D}\mathbf{T}^{\mathrm{T}}.$$

Thus the diagonal elements of **V** can be computed as a weighted sum of squares of the rows of **T**. Gauss gives two algorithms for computing **T**, one of them particularly advantageous when only a few variances are to be computed.

The second method is a very general result for computing the variance of an arbitrary linear combination

$$t = \mathbf{g}^{\mathrm{T}}\mathbf{b} + \kappa$$

of the unknowns **b**. Specifically, if we pass from the variables **b** to the variables **u**, so that t assumes the form

$$t = \mathbf{h}^{\mathrm{T}}\mathbf{u} + \hat{t},$$

then $\hat{t}$ is the value of t at the least squares estimates of the unknowns,[8] and its variance is proportional to

$$\mathbf{h}^{\mathrm{T}}\mathbf{D}\mathbf{h}.$$

Moreover, **h** may be obtained by solving the triangular system

$$\mathbf{R}^{\mathrm{T}}\mathbf{h} = \mathbf{g}.$$

Thus Gauss reduces the problem of computing a variance to that of solving a triangular system.

A modern practice in numerical linear algebra is to compute a matrix decomposition and then use it in a variety of computations. Although it would be anachronistic to call Gauss a decompositionalist, he calculated like one. The results of his elimination serve as a computational platform from which both estimates and variances can be obtained.

7. Computational Complexity

Did Gaussian elimination represent an improvement over the practices of the day? If we assume that people were using Gauss–Jordan elimination to solve systems, they would have performed roughly $\frac{1}{2}p^3$ multiplications and

[7] Laplace, for example, recommended a similar procedure in the first supplement to his *Théorie Analytique des Probabilités* [18].

[8] It has been asserted [22] that Gauss established that $\hat{t}$ enjoyed the same minimum variance properties as the components of $\hat{\mathbf{b}}$. Although the result is true, Gauss never proved it.

about the same number of additions. Gaussian elimination, on the other hand, requires about $\frac{1}{6}p^3$ multiplications and additions. Thus Gaussian elimination represents an improvement of a factor of about three.

If variances are required, the inversion of the normal equations by Gauss-Jordan elimination would cost an additional $\frac{1}{3}p^3$ multiplications and additions for a total of $\frac{5}{6}p^3$. With Gauss's approach the total is $\frac{1}{3}p^3$, an improvement by a factor $\frac{5}{2}$.

In an age in which a workstation can solve a system of order 100 with barely a hiccup, it is easy to be cavalier about factors of three. To see what it might have meant to people who had to do their calculations by hand, consider the following quote from *A Treatise on the Adjustment of Observations* published in 1884 by T. W. Wright [24, p. 173]:

> Dr. Hügel, of Hessen, Germany, states that he has solved 10 normal equations in from 10–12 hours, using a log. table, but that 29 equations took him seven weeks.

Without Gaussian elimination Dr. Hügel's twelve hours would have stretched to a day and a half, and his seven weeks to almost half a year.

8. Notation

Gauss, like most mathematicians of his time, made sparing use of subscripts and superscripts, prefering to use primes or sequences of letters to distinguish variables. For example, Gauss writes his linear model in the form

$$\begin{aligned} v &= ax + by + cx + \text{etc.} + l \\ v' &= a'x + b'y + c'x + \text{etc.} + l' \\ v'' &= a''x + b''y + c''x + \text{etc.} + l'' \text{ etc.} \end{aligned}$$

Here x, y, z, etc. are the unknowns we have been denoting by b_i and the v's are the errors. Although this expansive notation appears awkward to us, in Gauss's hands it could be quite expressive. For example, here (slightly edited) is how he writes the normal equations.

$$\begin{aligned} 0 &= [aa]x + [ab]y + [ac]z + \text{etc.} + [al] \\ 0 &= [ab]x + [bb]y + [bc]z + \text{etc.} + [bl] \\ 0 &= [ac]x + [bc]y + [cc]z + \text{etc.} + [cl] \text{ etc.} \end{aligned}$$

Note the elegant way in which the notation $[ab]$ suggests a sum of products from the a and b columns.

Gauss's notation for elimination is equally well considered. The following is from the *Supplementum* [8] to the *Theoria Combinationis*

$$\begin{aligned} [bb,1] &= [bb] - \tfrac{[ab]^2}{[aa]} \\ [bc,1] &= [bc] - \tfrac{[ab][ac]}{[aa]} \\ [bd,1] &= [bd] - \tfrac{[ab][ad]}{[aa]} \\ &\text{etc.} \\ [cc,2] &= [cc] - \tfrac{[ac]^2}{[aa]} - \tfrac{[bc,1]^2}{[bb,1]} \\ [cd,2] &= [cd] - \tfrac{[ac][ad]}{[aa]} - \tfrac{[bc,1][bd,1]}{[bb,1]} \\ &\text{etc.} \\ [dd,3] &= [dd] - \tfrac{[ad]^2}{[aa]} - \tfrac{[bd,1]^2}{[bb,1]} - \tfrac{[cd,2]^2}{[cc,1]} \end{aligned}$$

Here as above, a pair of letters indicates the position in the normal equations. The appended numerals indicate the level of elimination. Incidentally, this seems to be the first appearance of the inner product form of the algorithm, in which the matrix $\mathbf{R}$ is generated row by row. It is the preferred form for hand calculation, since one need only record an array of $\frac{1}{2}p^2$ numbers.

9. Legacy

The casting of Gauss's results in matrix notation in some sense trivializes them. With our knowledge of matrix algebra, we can leap ahead to results that researchers of Gauss's time could only arrive at by more pedestrian routes. Yet we must be careful not to be patronizing. Gauss and his successors accomplished a great deal with their techniques and notation.

For example, Gauss's presentation of his algorithm as elimination in a quadratic form strikes us as unusual today. Yet it was the first of many reductions of quadratic and bilinear forms that later became our familiar matrix decompositions, including among others the LU decomposition, the Jordan canonical form, and the singular value decomposition. As Kline points out in his book *Mathematical Thought from Ancient to Modern Times* [13, Ch. 33], by the time the use of matrices had become widespread, many of the principal results of matrix theory had already been established.

Gauss's algorithms, written in his notation, survived into the twentieth century, especially in books on geodesy. Thereafter, as people began to use present-day notation, his contributions became less visible. By 1959, when I first began working with computers, Gaussian elimination had come to mean any triangularization of a system of equations, symmetric or nonsymmetric, followed by a back substitution, and none of us had an idea of what Gauss had actually done.

Yet what he did is worth recalling. Gauss worked with real-life problems and got his hands dirty solving them. He always looked for the best, most efficient algorithm;

and when he had it, he expressed it in a clean notation that suggested how to use it. These virtues are no less important today than in Gauss's time.

References

[1] R. J. Boscovich and C. Maire. *De Litteraria Expeditione per Pontificiam ditionem ad dimetiendas duas Meridiani graduss.* Palladis, Rome, 1755. Cited in [23].

[2] C. F. Gauss. *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientium.* Perthes and Besser, Hamburg, 1809. Cited and reprinted in [11, v. 7, pp. 1–261]. English translation by C. H. Davis [10]. French and German translations of Book II, Part 3 in [9, 12].

[3] C. F. Gauss. Disquisitio de elementis ellipticis Palladis. *Commentatines societatis regiae scientarium Gottingensis recentiores*, 1, 1810. Cited and reprinted in [11, v. 6, pp. 1–64]. French translation of §§13–14 in [9]. German translation of §§10–15 in [12].

[4] C. F. Gauss. Anzeige: Theoria combinationis observationum erroribus minimis obnoxiae: Pars prior. *Göttingische gelehrte Anzeigen*, 33:321–327, 1821. Cited and reprinted in [11, v. 4, pp. 95–100].

[5] C. F. Gauss. Anzeige: Theoria combinationis observationum erroribus minimis obnoxiae: Pars posterior. *Göttingische gelehrte Anzeigen*, 32:313–318, 1823. Cited and reprinted in [11, v. 4, pp. 100–104].

[6] C. F. Gauss. Theoria combinationis observationum erroribus minimis obnoxiae: Pars posterior. *Commentatines societatis regiae scientarium Gottingensis recentiores*, 5, 1823. Cited and reprinted in [11, v. 4, pp. 27–53]. French and German translations in [9, 12].

[7] C. F. Gauss. Theoria combinationis observationum erroribus minimis obnoxiae: Pars prior. *Commentatines societatis regiae scientarium Gottingensis recentiores*, 5, 1823. Cited and reprinted in [11, v. 4, pp. 1–26]. French and German translations in [9, 12].

[8] C. F. Gauss. Supplementum theoriae combinationis observationum erroribus minimis obnoxiae. *Commentatines societatis regiae scientarium Gottingensis recentiores*, 6, 1828. Cited and reprinted in [11, v. 4, pp. 55–93]. French and German translations in [9, 12].

[9] C. F. Gauss. *Méthode des Moindres Carres.* Ballet–Bachelier, Paris, 1855. Translation by J. Bertrand of various works of Gauss on Least Squares.

[10] C. F. Gauss. *Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections.* Little, Brown, and Company, 1857. Translation by Charles Henry Davis of *Theoria Motus* [2]. Reprinted by Dover, New York, 1963.

[11] C. F. Gauss. *Werke.* Kiiglichen Gesellschaft der Wissenschaften zu Göttingen., 1870–1928.

[12] C. F. Gauss. *Abhandlungen zur Methode der kleinsten Quadrate.* P. Stankeiwica', Berlin, 1887. Translation by A. Borsch and P. Simon of various works of Gauss on Least Squares.

[13] M. Kline. *Mathematical Thought from Ancient to Modern Times.* Oxford University Press, New York, 1972.

[14] J.-L. Lagrange. Researches sur la métode de maximis et minimis. *Miscellanea Taurinensi*, 1, 1759. Cited and reprinted in [15, v. 1, pp. 1–16].

[15] J.-L. Lagrange. *Œvres de Langrange.* Gauthier–Villars, Paris, 1867–1892.

[16] P. S. Laplace. Sur quelques points du systéme du monde. *Memoires de l'Academie des Sciences de Paris*, 1789. Cited and reprinted in [19, v. 11, pp. 475–558].

[17] P. S. Laplace. Mémoire sur les intégrales définies et sur application aux probabilités. *Memoires de l'Academie des Sciences de Paris*, 11, 1810–1811. Cited and reprinted in [19, v. 12, pp. 355–412].

[18] P. S. Laplace. *Théorie Analytique des Probabilités.* Courcier, Paris, third edition, 1820. Reprinted in [19, v. 7].

[19] P. S. Laplace. *Œvres Compeétes.* Gauthier–Villars, Paris, 1878–1912.

[20] A. M. Legendre. *Nouvelle méthodes pour la détermination des orbites des comètes.* Courcier, Paris, 1805. Cited in [23], where the appendix on least squares is reproduced.

[21] R. L. Plackett. The discovery of the method of least squares. *Biometrika*, 59:239–251, 1972.

[22] H. L. Seal. The historical development of the Gauss linear model. *Biometrika*, 54:1–24, 1967.

[23] S. M. Stigler. *The History of Statistics.* Harvard University Press, Cambridge, Massachusetts, 1986.

[24] T. W. Wright. *A Treatise on the Adjustment of Observations.* Van Nostrand, New York, 1884.

Spatial Estimation and Presentation of Regression Surfaces in Several Variables Via the Averaged Shifted Histogram

Gerald Whittaker
ERS/USDA Room 937
1301 New York Ave., NW
Washington, D.C. 20005-4788

David W. Scott*
Department of Statistics
Rice University
Houston, TX 77251-1892

Abstract

A simple algorithm for estimating the regression function over the United States is introduced. The approach allows for data obtained from a complicated sampling design, as well as for the inclusion of a few additional covariates. The regression estimates are obtained from an associated probability density estimate, namely the averaged shifted histogram. The algorithm has proven especially successful over a large mesh, say 300 by 200 nodes, in a data rich setting, even on a 486 computer running Splus. Commonly available alternative codes including kriging failed to produce useful estimates in this setting.

1. Introduction

The problem of nonparametric regression has attracted a wealth of attention since the pioneering papers of Nadaraya (1964) and Watson (1964); see Eubank (1988) and Härdle (1990). Available algorithms range from the simple running median, to variational formulations giving rise to spline estimates, to kernel estimates, and finally local polynomial fitting. There has been a great deal of recent discussion about the right and wrong way to do nonparametric regression. Some have argued for the elegance of splines, while others find the local polynomial approach compelling, but some argue for one's personal preference.

From our experience in the density estimation setting, we find that direct methods work well in 1 to 5 dimensions, but even in 3–5 dimensions, the size of the meshes is growing exponentially, and sufficient data often aren't available. In the regression setting, we find that the discussion in the literature has focused too heavily on relatively small 1 and 2 dimensional data sets where most methods perform reasonably well. In this manuscript, we consider a more realistic and stressful problem dealing with farm data such as that routinely surveyed by the U.S.D.A. These surveys result in very large databases over nonuniform spatial meshes (see Figure 1), complicated by nonuniform weighting schemes as well as interest in several covariates.

Large data sets and/or large mesh sizes result in practical problems. Too many regression methods have solutions or algorithms whose exact form is determined by the number of data points (splines, kernels, etc.) that make computation infeasible even on 486 level computers. The key to computational efficiency is the same as for density estimation: **binning** the multivariate data (Scott, 1992; Härdle and Scott, 1992; Fan and Marron, 1994).

Beyond 4 or 5 dimensions, direct mesh methods of any kind encounter practical difficulties resulting from the curse of dimensionality. Some form of advanced projection technology or additive modeling has proven useful (Hastie and Tibsharani, 1990).

However, "real data" can throw a curve at the best planned evaluation of even carefully constructed algorithms. We have mentioned the special problem of large samples. Here we would like to focus on problems resulting from a mixture of spatial and continuous variables. They are: (1) irregular boundary definition, (2) data collected by a sampling design, and (3) a very large mesh required to have high spatial resolution. In principle, an exact irregular boundary scheme can be handled (perhaps with great programming effort), and weighting can be introduced into the estimation phase. However, many simple-minded implementations run into numerical instabilities with large meshes.

We wish to show how simple the binned methods (specifically the ASH or WARP algorithms) can be modified to handle such data, even with very fine 300 × 200 spatial meshes, on a 486 level machine.

We find that the common focus on boundary behavior is only a minor part of our thinking. Firstly, we are dealing with large samples and thus only a relatively small bandwidth is required. (By way of contrast, many simulation examples involve $n = 100$ 1-dimensional data where the bandwidth may span 1/4–1/2 of the data

*Supported in part by NSF grant DMS-9306658.

interval, making boundary conditions dominant.) Secondly, for mapping purposes, we find the boundary effects and corrections of little practical importance towards understanding and summarizing our data exploration/presentation efforts.

Ironically, we have found "internal boundary" situations more of a practical nuisance. These occur in areas internal to the USA, say, where there are no data (because there is no agriculture), inducing a boundary effect caused by sparseness rather than a physical external boundary. We identify this situation by observing how low the density falls in each region where we are evaluating the regression function (i.e., how close to 0 is the denominator?). This is a multivariate version of the well-known practical problem of "extrapolation" of regression estimates beyond the support of the data.

In our experience, many off-the-shelf kriging or regression programs cannot handle large rectangular meshes of 300 × 200 points covering a mercator projection of the lower 48 states. Rewriting such codes is always a possibility, but we have found that the simple ASH ideas provide excellent estimates and dramatic correlation with actual photographic evidence. Carr (1990) has used raw (hexagonal histogram) bivariate binning techniques. We are interested in providing some additional smoothing (that will provide improved estimation quality) as well as handling additional covariates.

2. Algorithm Motivation

We start with a simple description of the ideas and algorithms for handling (x, y, z) data where (x, y) represents the center of one of our bivariate bins (approximately 10 miles by 10 miles) containing one or more U.S.D.A. sampling units. The variable z represents the quantity of interest; for example, total farm income or the fraction of Federal dollars in farm income. We seek to estimate $\mathrm{E}\,[Z(x,y)]$ or $\bar{z}(x,y)$ in areas where $f(x,y) > 0$.

2.1. Kernel Regression Estimation

Let K be a symmetric kernel function with support on $(-1,1)$ satisfying $\int_{-1}^{1} K(t)\,dt = 1$. Given a positive smoothing parameter h, define the scaled kernel function by

$$K_h(t) = \frac{1}{h} K\left(\frac{t}{h}\right)$$

We take as a starting point the well-known result (Scott, 1992) that the Nadaraya-Watson bivariate regression estimator

$$\hat{m}(x,y) = \frac{\sum_{i=1}^{n} z_i K_h(x - x_i) K_h(y - y_i)}{\sum_{i=1}^{n} K_h(x - x_i) K_h(y - y_i)}$$

is the *exact* result of the computation

$$\hat{m}(x,y) = \int_z z\hat{f}(z|x,y)\,dz = \frac{\int z\hat{f}(x,y,z)\,dz}{\int \hat{f}(x,y,z)\,dz}$$

where the trivariate product kernel density estimator is given by

$$\hat{f}(x,y,z) = \frac{1}{n}\sum_{i=1}^{n} K_h(x-x_i)\,K_h(y-y_i)\,K_h(z-z_i)\,.$$

Clearly

$$\int \hat{f}(x,y,z)\,dz = \frac{1}{n}\sum_{i=1}^{n} K_h(x-x_i)\,K_h(y-y_i)$$

since $\int K_h(z-z_i)\,dz = \int K_h(z)\,dz = 1$.

Also, $\int z\hat{f}(x,y,z)\,dz = \sum z_i K_h(x-x_i)\,K_h(y-y_i)$, since

$$\int zK_h(z-z_i)\,dz = \int (z+z_i)K_h(z)\,dz = 0 + z_i\,,$$

recalling that $\int zK_h(z)\,dz = 0$ (by symmetry).

Clearly, different smoothing parameters h_x, h_y, h_z could be chosen for each dimension. Interestingly, the particular choice of h_z has no effect on the regression estimate!

It is well-known (Härdle, 1990) that local polynomial regression (LPR) estimators and spline methods have equivalent kernel forms. LPR does have the advantage that the kernel adjusts properly at the boundary to reduce bias (Fan, 1992).

However, the practical gain of the bias correction is often small, as $f(x) \to 0$ near the boundary and/or $m(x) \to 0$ near the boundary. Many authors consider only cases where $f(x)$ is nearly constant over a finite interval, or even the simplest case of a fixed equally-spaced mesh. These situations tend to accentuate boundary concerns and problems.

2.2. ASH Density Algorithm

We mimic the simple Nadaraya-Watson idea except on a more computationally oriented estimator, the averaged shifted histogram (ASH), introduced by Scott (1983, 1985, 1992). We remotivate the multivariate ASH.

Let us slightly alter our notation so that

$$x_1, x_2, \ldots, x_{n_x} \qquad y_1, y_2, \ldots, y_{n_y} \qquad z_1, z_2, \ldots, z_{n_z}$$

are the midpoints along each axis of a trivariate mesh of size $n_x \times n_y \times n_z$ with spacings $\delta_x, \delta_y, \delta_z$. Thus

$$\Delta x_i = \delta_x = \frac{h_x}{m_x} \quad \Delta y_i = \delta_y = \frac{h_y}{m_y} \quad \Delta z_i = \delta_z = \frac{h_z}{m_z}$$

for some integers m_x, m_y, m_z and smoothing parameters h_x, h_y, h_z.

Let ν_{jkl} denote the number of data points $(x,y,z)_i$ falling in bin B_{jkl}. Note that $\sum \nu_{jkl} = n$, and we expect many of the ν_{jkl} to be 0.

The "naive ASH" is constructed by "computing" $m_x \times m_y \times m_z$ (different) trivariate histograms, each with rectangular bin size $h_x \times h_y \times h_z$, but with origins shifted by multiples of $\delta_x, \delta_y, \delta_z$ along the coordinate axes. To be specific, one bin is anchored at the point $(j\delta_x, k\delta_y, l\delta_z)$, as j,k,l each range from 0 to $n_x - 1, n_y - 1, n_z - 1$.

Scott (1985) showed that this was a special case of a general weighting scheme:

$$\hat f_{jkl} = \hat f(x_j, y_k, z_l) = \frac{1}{n\delta_x\delta_y\delta_z}\sum_{a,b,c} w_{abc}\nu_{j+a,k+b,l+c}$$

where the sums range over $-m_x < a < m_x$, $-m_y < b < m_y$, and $-m_z < c < m_z$, and

$$w_{abc} = \frac{K\left(\frac{a}{m_x}\right)K\left(\frac{b}{m_y}\right)K\left(\frac{c}{m_z}\right)}{\sum_a\sum_b\sum_c K\left(\frac{a}{m_x}\right)K\left(\frac{b}{m_y}\right)K\left(\frac{c}{m_z}\right)},$$

where K is supported on $(-1,1)$ as before. Note that in an obvious notation, $w_{abc} = w_a w_b w_c$. This is a classic discretization scheme. The weights $\{w_a, w_b, w_c\}$ need only be computed once.

We first verify that the trivariate ASH is indeed a density function. Clearly it is nonnegative. To prove that it has integral 1, we compute

$$\int\int\int \hat f(x,y,z)\,dx\,dy\,dz = \delta_x\delta_y\delta_z\sum_j\sum_k\sum_l \hat f_{jkl}$$

$$= \sum_j\sum_k\sum_l \frac{1}{n}\sum_a\sum_b\sum_c w_{abc}\nu_{j+a,k+b,l+c}$$

$$= \frac{1}{n}\sum_a\sum_b\sum_c w_{abc}\sum_j\sum_k\sum_l \nu_{j+a,k+b,l+c}$$

$$= \sum_a\sum_b\sum_c w_{abc} = 1\,,$$

assuming a buffer of 0's around the edges of the $\{\nu_{jkl}\}$ array, so that

$$\sum_j\sum_k\sum_l \nu_{j+a,k+b,l+c} = 1 \quad \text{for all } a,b,c.$$

In practice, the array $\{\hat f_{jkl}\}$ is initialized to all 0's, and then the influence of every bin B_{jkl} for which $\nu_{jkl} > 0$ is added to the appropriate subset of $\hat f_{jkl}$.

We could define $\hat f(x,y,z)$ to be a spline surface interpolated from the above array, but for simplicity, we take it to be constant over each bin B_{jkl} and assume it vanishes outside the mesh; that is, $\hat f(x,y,z) = 0$ there.

2.3. ASH Regression Algorithm

Following the Nadaraya-Watson motivation, the ASH regression estimator is found by computing

$$\hat m_{jk} = \hat m(x_j, y_k) = \mathrm{E}(Z|X = x_j, Y = y_k)$$

$$= \int z\hat f(z|x_j,y_k)\,dz = \frac{\int z\hat f(x_j,y_k,z)\,dz}{\hat f(x_j,y_k)}.$$

The numerator can be computed by integrating bin by bin along the z axis:

$$\sum_{l=1}^{n_z}\int_{z_l-\delta_z/2}^{z_l+\delta_z/2} z\hat f(x_j,y_k,z=z_l)\,dz = \sum_{l=1}^{n_z}\delta_z z_l \hat f(x_j,y_k,z_l)\,,$$

since $\int z\,dz = \delta_z z_l$ for the limits given (recall $\hat f$ is constant over each bin). Thus

$$\hat m_{jk} = \frac{\sum_l \delta_z z_l\left\{\frac{1}{n\delta_x\delta_y\delta_z}\sum_a\sum_b\sum_c w_{abc}\nu_{j+a,k+b,l+c}\right\}}{\frac{1}{n\delta_x\delta_y}\sum_a\sum_b w_{ab}\nu_{j+a,k+b}}$$

$$= \frac{\sum_a\sum_b w_{ab}\sum_c w_c\sum_{l=1}^{n_z} z_l\nu_{j+a,k+b,l+c}}{\sum_a\sum_b w_{ab}\nu_{j+a,k+b}}.$$

Now the final sum in the numerator can be computed by observing that it is almost a conditional expectation:

$$\sum_{l=1}^{n_z} z_l\frac{\nu_{j+a,k+b,l+c}}{\nu_{j+a,k+b}}\cdot\nu_{j+a,k+b} = \bar z_{ab}\,\nu_{j+a,k+b}$$

as we let $m_z \to \infty$ (or equivalently let $\delta_z \to 0$ with h_z fixed), where

$$\bar z_{ab} = \frac{1}{n_{ab}}\sum_{(x,y,z)_i\in B_{ab}} z_i\,.$$

Continuing, we note that $\sum w_c = 1$, so that we finally arrive at the final form of the ASH regression estimator as:

$$\hat m_{jk} = \frac{\sum_a\sum_b \bar z_{ab}w_{ab}\nu_{j+a,k+b}}{\sum_a\sum_b w_{ab}\nu_{j+a,k+b}}.$$

2.4. ASH Regression Extensions

REMARK 1: For the survey sampled data, each data point takes the extended form

$$\{(x,y,z,\alpha)_i\,,\quad i = 1,\ldots,n\}\,,$$

where α_i is the effective sampling weight. Previously, we have assumed that $\alpha_i = 1$ for all cases. Here, the frequency counts ν_{jkl} are replaced by the sum of these α_i weights rather than 1's.

REMARK 2: Occasionally, our data will include other covariates and be of the form

$$\{(x,y,z,t,\alpha)_i\,,\quad i=1,\ldots,n\}\ ,$$

where t is some covariate of interest. Then we compute the ASH regression estimator $\hat{m}(x,y,t)$ by simply adding another loop to the numerator and denominator of the $\hat{m}_{jk}$ equation above. The sampling weights are the same of course. What could be easier? Typically, we will map the estimate at several levels of t, for example, $\hat{m}(x,y,t=t_0)$.

REMARK 3: The 1-dimensional ASH regression prescription was first published in Härdle and Scott (1992) under the name WARPing.

3. Mapping Details

After the "usa()" is plotted, the regression ASH is computed over the entire 300 × 200 mesh and added to the figure by using either the Splus "contour" or "image" function and the argument "add=T". Typically, the contour lines will extend slightly outside the US borders. A simple trick removes those lines, by applying "polygon" to two pieces that outline half the borders of the US and the surrounding rectangles. This will be illustrated in the examples.

The internal boundary solution is not handled in an elegant fashion currently. Thresholding could be applied, but we find the problem is relatively localized and have left it for the reader to discover. A bootstrap algorithm has been implemented to estimate the pointwise error. We have used this to replace or delete regions where the estimator behaves erratically.

4. Examples

The "real" data considered in this section come from the Farm Costs and Returns Survey. This is a stratified complex design survey which is used to measure finances and production of all U.S. agriculture. The weight of each observation was taken to be the inverse of the probability of selection. We begin with a small bivariate simulation.

4.1. A Simulation Example

A surface with 3 bumps typical of those encountered in USDA work was constructed on a 50 × 50 mesh (not shown). The surface was contaminated twice: first with Gaussian noise and then with Cauchy noise. From this complete set of 2,500 points, 200 points were selected at random. The estimated ASH regression surface was computed with $m_x = m_y = 5$. The trimodal structure was evident, but then so were some spurious peaks induced by the Cauchy noise. Clearly, the raw ASH algorithm has no robustness component included.

We next applied the loess (Cleveland, 1979) Splus function to these data. A coplot of x vs. z given y was computed and a perspective plot of the entire estimated surface examined. The loess surface is significantly better as it includes iteration to provide more robust answers to minimize the effects of the Cauchy noise.

4.2. Farm Costs and Returns Survey Example

A sample of n = 13,000 of 1.7 million farms was drawn. For these data, the FIPS code for each observation was known. Thus the exact location of each observation was assigned to the location of the population centroid of the county where the farm is located. The map of the 3,100 centroids is shown in Figure 1. Observe that the resolution is much greater east of the Mississippi.

When loess, kriging, and other methods were applied to these data, each failed to produce a usable surface from the data. The result was always a smooth surface for most of the country with an enormous peak at an edge. However, the ASH regression algorithm with $m_x = m_y = 5$ produced excellent results.

We first computed the estimate *without using the sampling weights* as shown in Figure 2, while the estimate with sampling weights is shown in Figure 3. This made a big difference, particularly in areas where there are many observations with small weights.

As mentioned earlier, internal boundaries can cause problems for the algorithm. In Figure 4, we zoom in on one of the problem areas. The four corners region of the Southwest (Utah, Arizona, Colorado, and New Mexico) join at about the location where this peak occurs. The surface rises gradually to the peak, becomes a flat plateau, then drops off a cliff to an area of no data (where the regression estimator becomes 0/0). Use of zipcode centroids and adaptive bandwidths might solve this.

Figure 5 captures our final estimate of the fraction of government payments to gross farm income. Note the contours are shown on a logarithmic scale. The boundary artifact in the four corners region can be searched out. Otherwise, no other glaring boundary problems appear. For the most part, the value of the regression surface is quite small near the US borders, except in Texas and along a portion of the border with Canada (where government subsidies are even greater!). We do not find the bias incurred particularly misleading.

Next, we included a surrogate variable t to capture the "size" of each farm. This was simply the total sales. We computed $\hat{m}(x,y,t)$ using the extended ASH algorithm and computed 2 slices—one for small farms (Figure 6) and one for large farms (Figure 7). The highest subsidies for small farms are concentrated primarily in the Midwest and Plains states. For large farms, we see the rice farms along the Mississippi, for example. These patterns are quite interesting to policy makers.

4.3. Overlaying Maps

A popular exercise is overlaying different maps to capture a relationship. Conventionally, this is done following county boundaries. For example, Figure 8 displays such data. The viewer is required to form a "mental surface" or internal representation for these data. The ASH algorithm does this for the viewer, with the added advantages of consistency and the application of objective statistical criterion to decide the contours of the surfaces. In Figure 9, 4 shades are indicated on the map coming from 2 ASH estimates. White areas indicate low activity on both scales. The darkest shaded areas indicate where both (1) farms are dependent on government payments and (2) the geographical areas are highly dependent on farm income. Such information is more easily gleaned from these smooth ASA estimates.

5. Discussion

The naive ASH is not robust, but is easily adapted to handle weighted data and covariates with small computational overhead. Elegant procedures without covariate handling have been considered by Tobler (1979). We have not taken advantage of possible small gains available by considering spatial correlations.

However, kriging and lowess both produced estimates with huge values at the boundary and outside the US borders. Apparently, the trick of placing a rectangular grid on the US extending outside the borders fails because the algorithms require explicit knowledge of the boundary locations as input.

The actual proximate reason for failure, interestingly enough, is due to the "adaptive" nature of these algorithms, which fit the LPR over a region with a certain fraction of the data. In places where the mesh extends offshore, the regression estimate is reaching far inland for any data to fit — the extrapolation problem once again. (Explicit boundary handling would fix this, presumably).

The ASH procedure used a fixed (or nonadaptive) neighborhood. The result is regions where the regression estimate is undefined (0/0). However, we are more comfortable with such undefined regions than with providing dubious estimates obtained by spanning empty spaces.

6. References

Carr, D.B. (1990). "Looking at Large Data Sets Using Binned Data Plots," PNL-7301, Battelle Labs, Richland, WA.

Cleveland, W.S. (1979). "Robust Locally Weighted Regression and Smoothing Scatterplots." *Journal of the American Statistical Association*, 87:829–836.

Eubank, R.L. (1988). *Spline Smoothing and Nonparametric Regression.* Marcel Dekker, New York.

Fan, J. (1992). "Design Adaptive Nonparametric Regression." *Journal of the American Statistical Association*, 87:998–1004.

Fan, J. and Marron, J.S. (1994). "Fast Implementations of Nonparametric Curve Estimators." *Journal of Computational and Graphical Statistics*, 3:35–56.

Härdle, W. (1990). *Applied Nonparametric Regression.* Cambridge University Press, Boston.

Härdle, W. and Scott, D.W. (1992). "Smoothing by Weighted Averaging of Shifted Points," *Computational Statistics*, 7:97–128.

Hastie, T. and Tibsharani, R. (1990). *Generalized Additive Models.* Chapman and Hall, London.

Nadaraya, E.A. (1964). "On Estimating Regression," *Theory of Probability and its Applications*, 9:141–142.

Watson, G.S. (1964). "Smooth Regression Analysis," *Sankhy A*, 26:359–372.

Scott, D.W. (1983). "Nonparametric Probability Density Estimation for Data Analysis in Several Dimensions," *Proceedings of the Twenty-Eighth Conference on the Design of Experiments in Army Research Development and Testing*, pp. 387–397.

Scott, D.W. (1985). "Averaged Shifted Histograms: Effective Nonparametric Density Estimators in Several Dimensions," *Annals of Statististics*, 13:1024–1040.

Scott, D.W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, New York.

Tobler, W.R. (1979). "Smooth Pycnophylactic Interpolation for Geographical Regions," *Journal of the American Statistical Association*, 74:519–530.

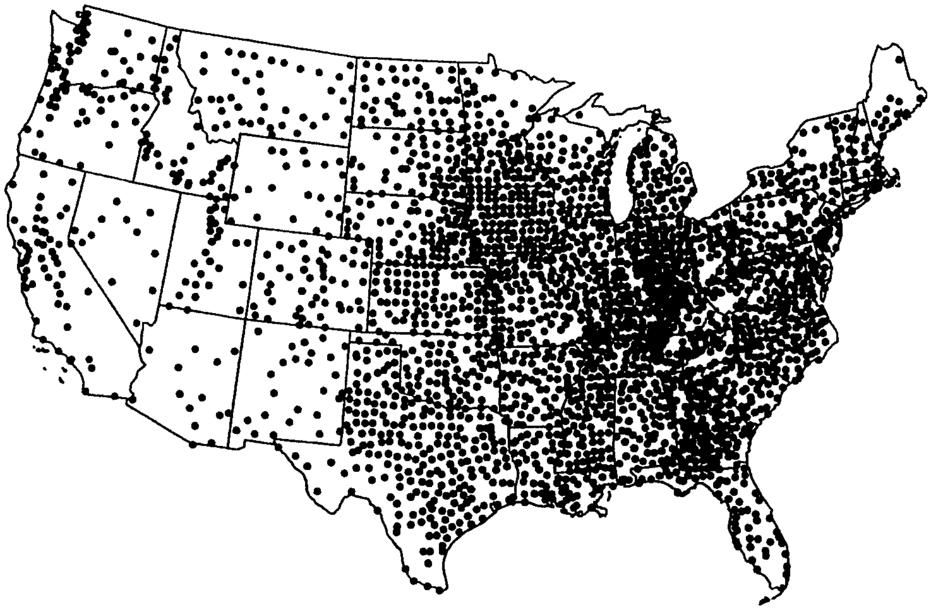


Figure 1. Population centroids of all U.S. counties.

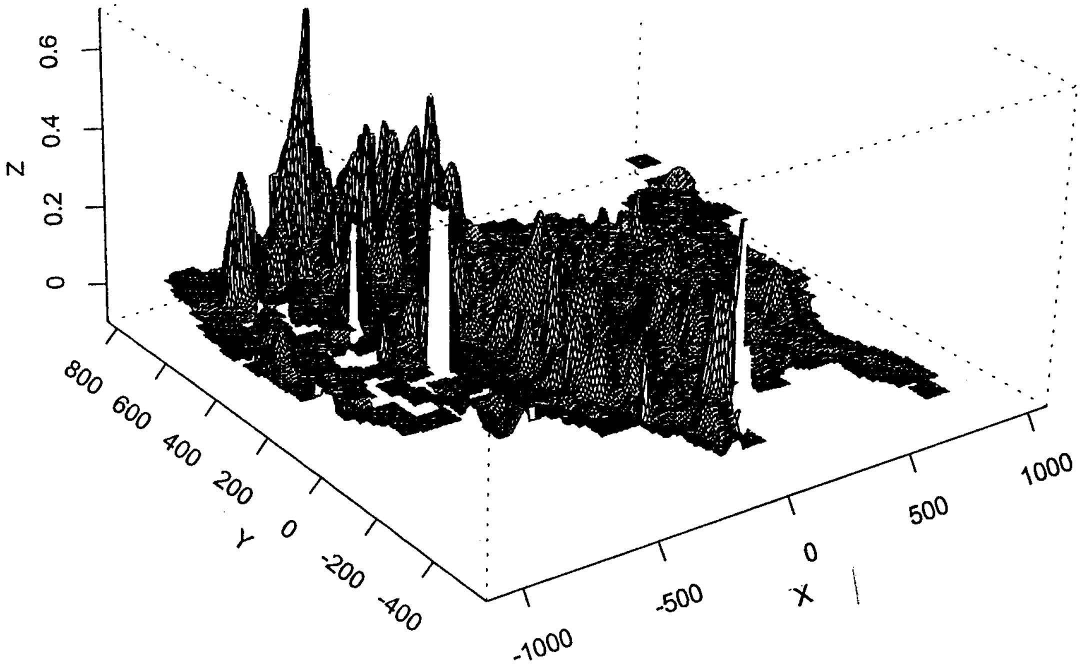


Figure 2. ASH estimates with equal weights $\alpha_i = 1$. Note the low values east of the Mississippi River.

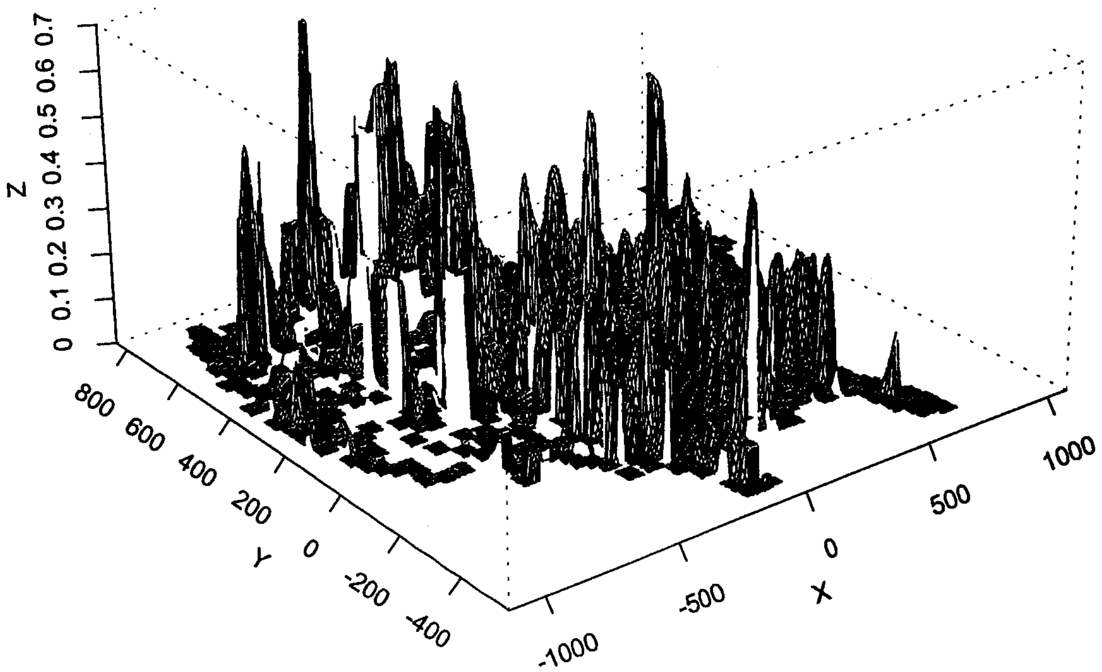


Figure 3. Correctly weighted ASH estimate compared to Figure 2.

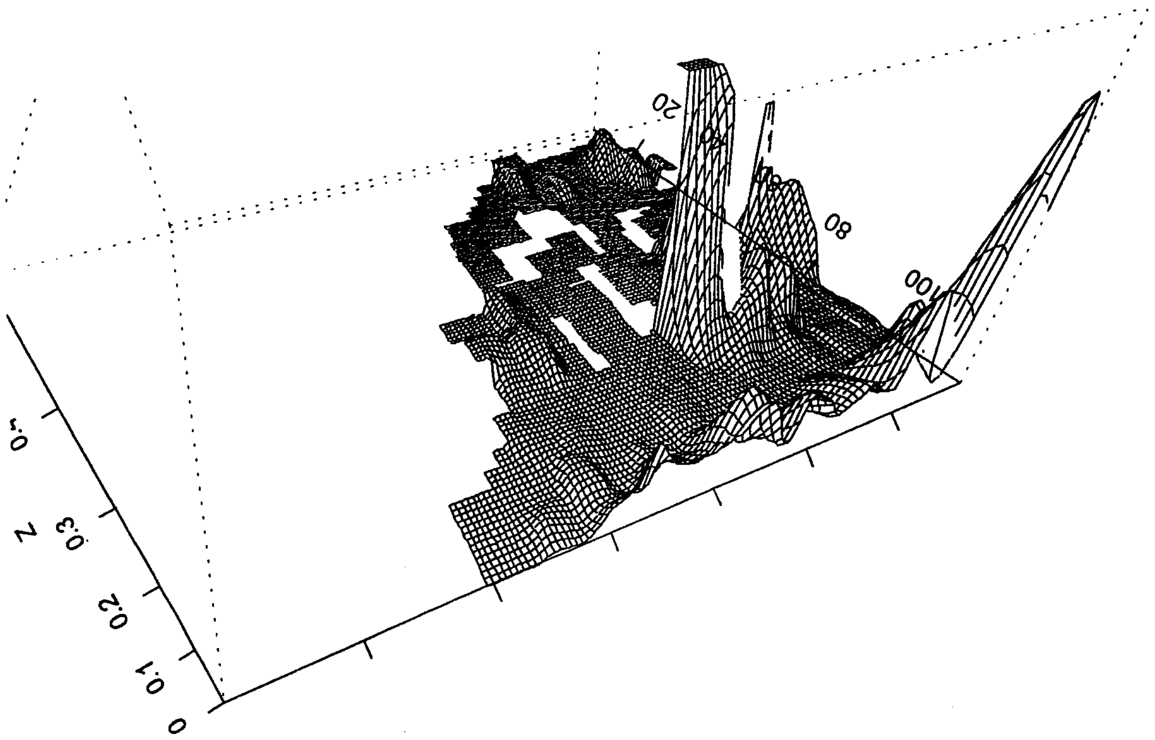


Figure 4. Blowup of ASH estimate near an internal boundary.

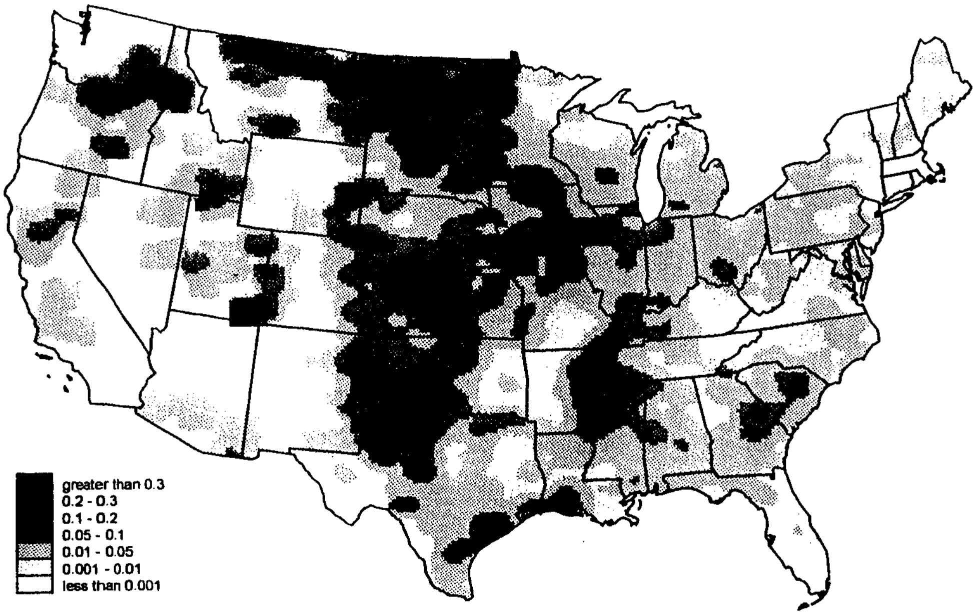


Figure 5. Contours of the proportion of farm income from Federal estimated by the ASH.

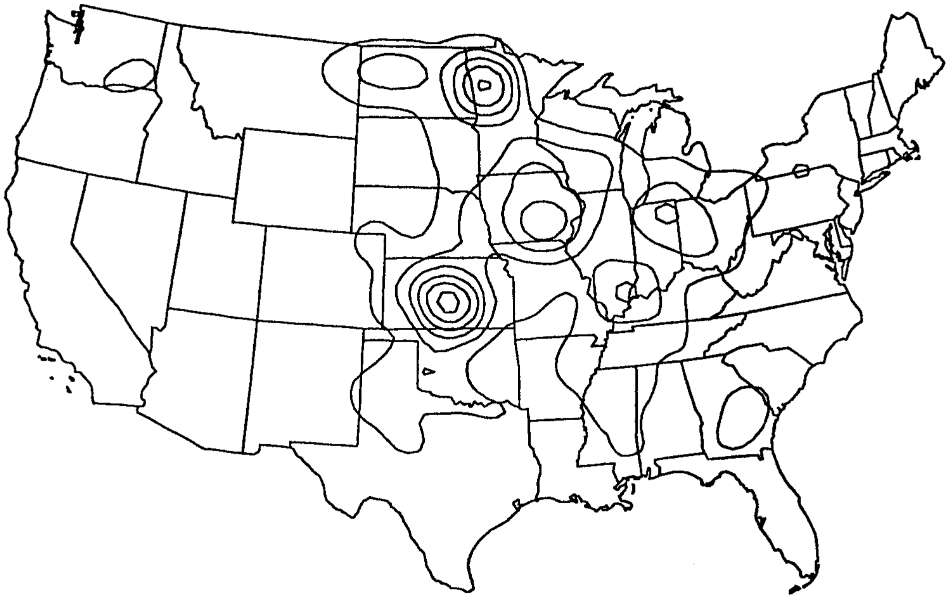


Figure 6. Conditional distribution of the variable in Figure 5 for "small" farms.

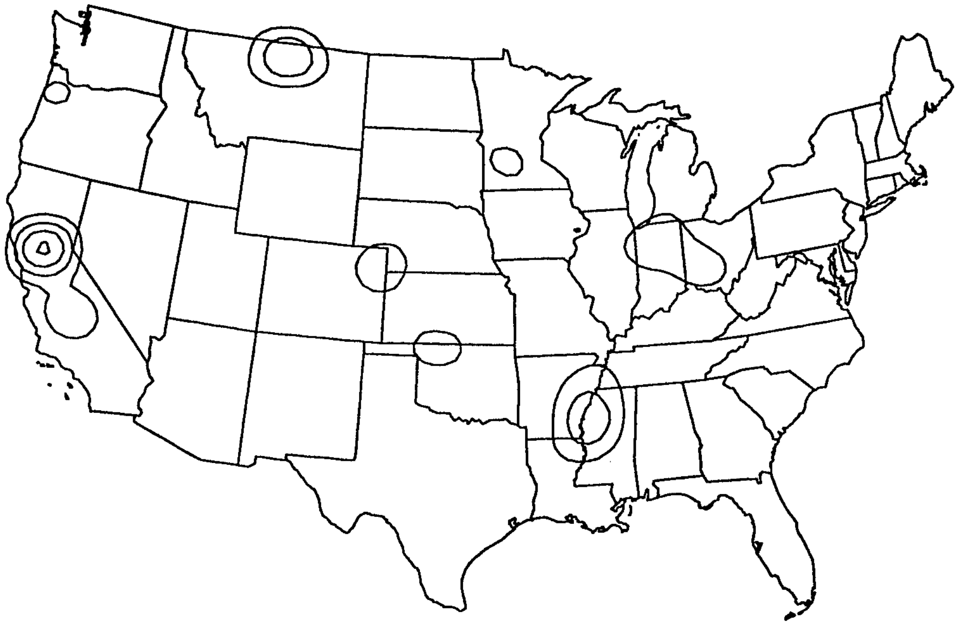


Figure 7. Data as in Figure 6 but for "large" farms.

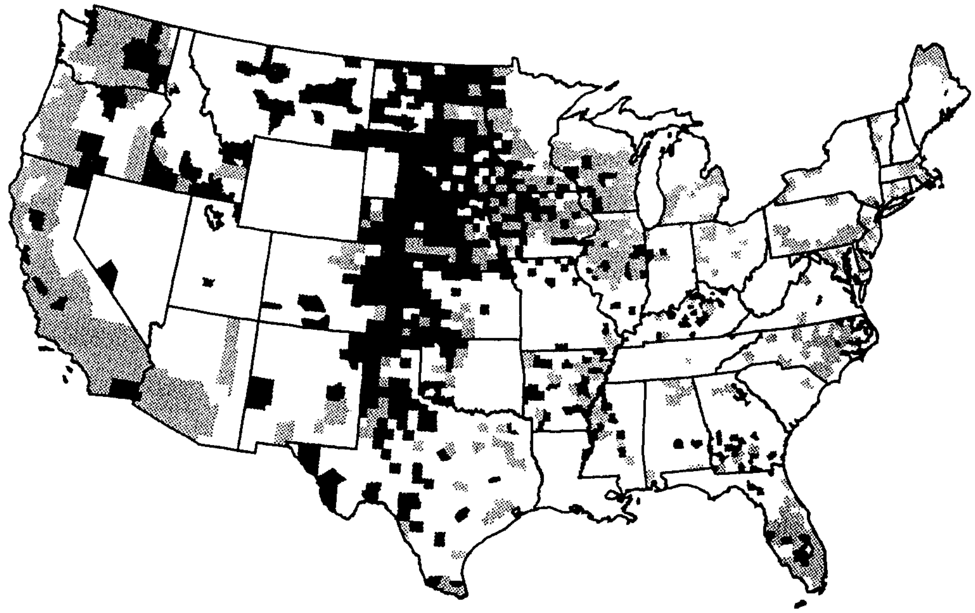


Figure 8. Data presented in the usual fashion, on county-by-county basis.

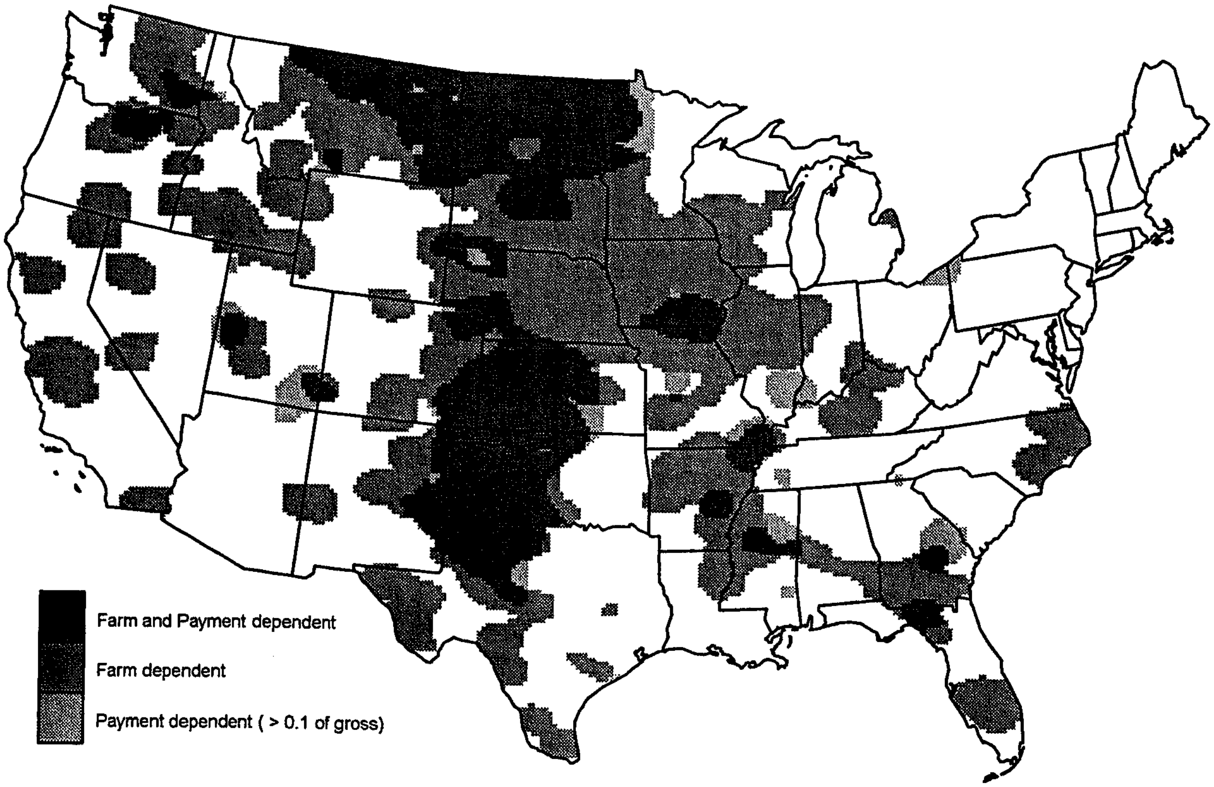

Farm and Payment dependent
Farm dependent
Payment dependent (> 0.1 of gross)

Figure 9. Overlay of 2 ASH regression estimates (see text).

Fast and Stable Computation of Local Polynomials

Burkhardt Seifert
Abteilung Biostatistik
Universität Zürich
Sumatrastrasse 30
CH–8006 Zürich

Michael Brockmann
Universität Heidelberg
Im Neuenheimer Feld 294
D–6900 Heidelberg

Joachim Engel
Wirtschaftstheoretische Abt. II
Universität Bonn
Adenauerallee 24–26
D–5300 Bonn

Theo Gasser
Abteilung Biostatistik
Universität Zürich
Sumatrastrasse 30
CH–8006 Zürich

Abstract

Naive implementations of local polynomial fits require almost $O(n^2)$ operations. In this paper a fast $O(n)$–algorithm is presented. It is based on updating normal equations. Numerical stability is guaranteed by centering while moving, controlling ill–conditioned situations for small bandwidths and data–tuned restarting the updating procedure. "Exact binning" and restarting at every output point results in a moderately fast but highly stable $O(n^{7/5})$ algorithm. Applicability of algorithms is evaluated for estimation of regression curves and their derivatives.

Some key words: Fast computation; Local polynomials; Nonparametric estimation; Nonparametric regression; Smoothing; Updating.

AMS 1991 subject classification. Primary 65D10, Secondary 62G07, 65D25.

1 Introduction

Nonparametric methods of curve estimation have become useful techniques. For applications fast algorithms which allow computation on personal computers and at the same time guarantee numerical stability are highly desirable. In particular, when choosing the bandwidth from the data or in bootstrapping schemes, multiple evaluations of the estimators become necessary and a fast algorithm is even more desirable. Furthermore, due to technical progress, automatic recording of mass data has become easier. This puts higher demands on statistical algorithms.

For various spline based regression estimators algorithms have been developed whose number of arithmetic operations grows only linearly with the number of data points n (see de Boor, 1978; Utréras, 1980, 1981; Silverman, 1984; Hutchinson & de Hoog, 1985). In contrast, a naive implementation of a kernel estimator for regression or density estimation requires almost $O(n^2)$ operations. Through averaging shifted histograms Scott (1985, 1986) proposed a fast density estimator approximating a kernel estimator which needs $O(n)$ operations. Härdle & Scott (1992) extended this idea through their concept of WARPING (weighted average of rounded points) to the regression case where their estimator approximates the Nadaraya–Watson kernel estimator. A fast algorithm for an exact convolution type kernel regression was suggested by Gasser & Kneip (1989). Seifert, Brockmann, Engel & Gasser (1994) presented two fast $O(n)$ algorithms and a highly stable but slightly slower $O(n^{7/5})$ version of the latter algorithm. The algorithms are applicable to local polynomial regression and to kernel estimation.

This paper is based on Seifert et al. (1994). In section 2 the local polynomial regression estimator is briefly discussed. In section 3 a fast algorithm is derived. Its speed is based on updating normal equations and the idea of exact binning. Stability is obtained by several steps, centering while moving, control of ill–conditioned matrices and data–tuned restart of the updating procedure being the most important ones. Restarting at every output point results in a moderately fast $O(n^{7/5})$ algorithm, which is even more stable than the conventional one. A numerical evaluation is given in section 4 for estimation of regression curves and their derivatives in fixed and random designs.

2 Local Polynomial Regression

Let $(X_1, Y_1), \ldots, (X_n, Y_n)$ be a set of independent and identically distributed pairs of random variables where the X_i are scalar predictors and the Y_i are scalar responses. The developments of this paper can, however,

[1] This work was part of the research program no. 21.–36042.92 of the swiss NSF

be generalized to higher–dimensional design.

In regression analysis a functional relationship between predictor and response is assumed as

$$r(x) \quad = \quad \mathrm{E}(Y \mid X = x)\,. \qquad (1)$$

Predictors following a fixed design can be treated similarly. The predictors are assumed to be sorted $X_1 \leq \ldots \leq X_n$. The goal is to estimate $r(x_0)$ or its ν–th derivative $r^{(\nu)}(x_0) = \frac{d^\nu}{dx^\nu} r(x)\big|_{x=x_0}$ for some ν. The local polynomial approach is based on the approximation

$$r(x) \quad \approx \quad \sum_{j=0}^{p} \frac{r^{(j)}(x_1)}{j!}\,(x - x_1)^j \qquad (2)$$

provided x is close to x_1, where r is at least $(p+1)$ times differentiable. This representation suggests minimizing

$$\sum_{i=1}^{n} \left(Y_i - \sum_{j=0}^{p} \beta_j\,(X_i - x_1)^j \right)^2 K\left(\frac{X_i - x_0}{h}\right) \qquad (3)$$

with respect to $\beta = (\beta_0, \ldots, \beta_p)'$. Here K denotes a positive and symmetric weight function and h is the bandwidth. Denote

$$X \quad = \quad \begin{pmatrix} 1 & (X_1 - x_1) & \ldots & (X_1 - x_1)^p \\ \vdots & \vdots & & \vdots \\ 1 & (X_n - x_1) & \ldots & (X_n - x_1)^p \end{pmatrix}_{n \times (p+1)},$$

$$Y \quad = \quad \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix},$$

$$W \quad = \quad \mathrm{diag}(K\left(\frac{X_1 - x_0}{h}\right), \ldots, K\left(\frac{X_n - x_0}{h}\right)),$$

$$S_n \quad = \quad X'\,W\,X = \begin{pmatrix} S_{n,0} & \ldots & S_{n,p} \\ \vdots & & \vdots \\ S_{n,p} & \ldots & S_{n,2p} \end{pmatrix}, \qquad (4)$$

and

$$T_n \quad = \quad X'\,W\,Y = \begin{pmatrix} T_{n,0} \\ \vdots \\ T_{n,p} \end{pmatrix} \qquad (5)$$

Then the solution of the least squares problem (3) is obtained as solution $\hat\beta$ of the linear system

$$S_n\,\hat\beta \quad = \quad T_n\,. \qquad (6)$$

The resulting local polynomial $\sum_{j=0}^{p} \hat\beta_j (x - x_1)^j$ is independent of x_1. We estimate the ν–th derivative of r at point x_0 by

$$\hat r^{(\nu)}(x_0) = \nu! \sum_{k=\nu}^{p} \binom{k}{\nu} (x_0 - x_1)^{k-\nu}\,\hat\beta_k\,. \qquad (7)$$

We assume, that X has full rank, i.e. that there are at least $p+1$ points in the local smoothing interval. Then $\hat r^{(\nu)}(x_0)$ is unique. Algorithmically this is achieved by increasing the bandwidth locally until $p+1$ points fall in the interval.

Asymptotic properties are studied in Fan (1993), Ruppert & Wand (1992) and Fan et al. (1993). In the latter it is shown that $\hat r^{(\nu)}(x_0)$ is an asymptotically minimax efficient estimator among all linear estimators. The Epanechnikov weight function $K(x) = (3/4)\,(1 - x^2)_+$ is optimal for estimating the regression function r itself, as well as its ν–th order derivative. Note that $p - \nu > 0$ should be odd according to asymptotic theory, and that usually $p - \nu$ is equal to 1 or at most 3 due to the local nature of the approximation. The local polynomial method automatically adapts to the boundary; the equivalent kernel is a boundary kernel as defined by Gasser et al. (1985). This feature of the local polynomial method saves extra computations at boundary points.

3 Algorithms for Local Polynomial Fitting

3.1 The conventional algorithm

Using $x_1 = x_0$ we have

$$S_{n,j} \quad = \quad \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)(X_i - x_0)^j, \qquad (8)$$

$$T_{n,j} \quad = \quad \sum_{i=1}^{n} K\left(\frac{X_i - x_0}{h}\right)(X_i - x_0)^j\,Y_i\,. \qquad (9)$$

Thus, finite moments with respect to the design points essentially determine the local polynomial fit. This is also true for higher dimensional design.

Once S_n and T_n have been computed, the local polynomial fit is obtained by solving the linear system (6). The computational effort is independent of n. (We will approach the problem of solving the normal equations later on, using the Cholesky decomposition.) Hence a fast algorithm relies on the fast computation of S_n and T_n over the entire output grid.

Usually, the output grid will consist of n points as the input grid (e.g. for cross–validation), or of a fraction of n, if n is large, or a multiple of n, if n is small (e.g. for graphical representation). If the number of points in the output grid is thus $m = O(n)$, then a conventional implementation of (6) will require $O(n^2 h)$ operations, based on weight functions with compact support. For standard regression estimation the optimal h is of order $O(n^{-1/5})$, leading to $O(n^{9/5})$ operations for a curve fit. However, for small bandwidths $h = O(n^{-1})$ (when the estimator is close to interpolation) the conventional implementation approaches $O(n)$ operations.

Now we derive fast algorithms, based on a polynomial weight function

$$K(x) = (\sum_{k=0}^{a} a_k\, x^k)\, \mathrm{I}_{[-1,+1]}(x) \qquad (10)$$

comprising in particular the optimal Epanechnikov $K(x) = (3/4)\,(1-x^2)_+$ and the minimum variance (uniform) weights. For simplicity we will present the algorithms only for S_n, since the computation of T_n is then straightforward. Moreover, we present the case of a constant or global bandwidth h, but in fact our algorithms work for local bandwidths $h = h(x_0)$ as well.

3.2 A "naive" fast algorithm — the idea of updating

Using the binomial formula in (8) and rearranging summation we get

$$\begin{aligned}
S_{n,j} &\\
&= \sum_{i=1}^{n} \left(\sum_{k=0}^{a} a_k \left(\frac{X_i - x_0}{h} \right)^k \right) \mathrm{I}_{[x_0-h,x_0+h]}(X_i)\,(X_i - x_0)^j \\
&= \sum_{k=0}^{a} h^{-k}\, a_k \sum_{i=1}^{n} (X_i - x_0)^{j+k}\, \mathrm{I}_{[x_0-h,x_0+h]}(X_i) \qquad (11) \\
&= \sum_{k=0}^{a} h^{-k}\, a_k \sum_{\ell=0}^{j+k} \binom{j+k}{\ell} (-x_0)^{j+k-\ell} \\
&\quad \times \left\{ \sum_{i=1}^{n} X_i^{\ell}\, \mathrm{I}_{[x_0-h,x_0+h]}(X_i) \right\} \qquad (12)
\end{aligned}$$

Given the value of $S_{n,j}$ at x_0, we can save a lot of computations by reusing the inner sums (in braces) over i when calculating $S_{n,j}$ at the next output grid point x_{01} say. From the inner sum we subtract the terms that are not in $[x_{01}-h, x_{01}+h]$, and add those terms which are in this interval, but do not belong to $[x_0 - h, x_0 + h]$. This results in a fast $O(n)$ algorithm, which is reminiscent of the old add/subtract box car smoothing (compare e.g. Eddy, 1980). Independent of h and j one has to calculate the terms X_i^{ℓ}, $i = 1, \ldots, n$, $0 \le \ell \le 2p + a$ only once. However, this algorithm is numerically instable. The main source of instability is the expansion of the term $(X_i - x_0)^{j+k}$. The add/subtract idea then leads to an accumulation of numerical errors. The problem is comparable to the well known instability of the textbook one–pass algorithm for estimation of a variance.

One way out is the use of centered quantities $(X_i - x_0)^k$ only, or quantities centered by $\bar{X}_0$, the mean of design points in the interval $[x_0-h, x_0+h]$ (as is common use in polynomial regression and done in this paper).

If we move towards the boundary, an increasing numerical instability is expected and observed. Then typically the number of points in $[x_0 - h, x_0 + h]$ decreases, which leads to smaller quantities $S_{n,j}$, and hence increasing relative numerical errors. Also, the weights at the boundary become larger by order of magnitude. This difficulty is dealt with by running from both ends to the middle of the estimation interval.

Our goals are the following: We would like to have a fast and stable algorithm over the entire domain of bandwidths, starting with an h containing the minimal number of design points which is $p + 1$ and going up to the maximal h. Numerical stability should be guaranteed for the Epanechnikov and the uniform weight function, i.e. $a = 2$ and $a = 0$. Of interest are the regression function itself ($\nu = 0$) and the first and second derivative ($\nu = 1, 2$), whereas $\nu = 3, 4$ might be needed for estimating smooth functionals only, e.g. for selecting optimal bandwidths (Gasser et al., 1991). Usually, we are satisfied to use a polynomial of order $p = \nu + 1$, but for $\nu = 0, 1, 2$ the choice of higher order p also may be of interest.

The above algorithmic steps will not be sufficient to reach these goals. The most important additional techniques consist of detecting ill–conditioned cases for small bandwidths and automatic restarting the updating procedure, based on properties of the computed matrix S_n (see section 3.4 below).

3.3 A fast and stable algorithm — the idea of centering while moving

To avoid numerical instability of the naive fast algorithm based on (12), it is necessary to use centered quantities only. In Seifert et al. (1994) two stable algorithms using $x_1 = x_0$ and

$$x_1 = \bar{X}_0 = \frac{\sum_{i=1}^{n} X_i\, \mathrm{I}_{[x_0-h,x_0+h]}(X_i)}{\sum_{i=1}^{n} \mathrm{I}_{[x_0-h,x_0+h]}(X_i)}$$

were presented. Here, we present a fast algorithm using $x_1 = \bar{X}_0$, the mean of design points to be used for estimation of $r^{(\nu)}(x_0)$. Then

$$\begin{aligned}
&S_{n,j}\\
&= \sum_{i=1}^{n} \left(\sum_{k=0}^{a} a_k \left(\frac{X_i - x_0}{h} \right)^k \right) \mathrm{I}_{[x_0-h,x_0+h]}(X_i)\,(X_i - \bar{X}_0)^j\\
&= \sum_{i=1}^{n} \sum_{k=0}^{a} h^{-k}\, a_k\\
&\quad \times \sum_{\ell=0}^{k} \binom{k}{\ell} (X_i - \bar{X}_0)^{j+\ell} (\bar{X}_0 - x_0)^{k-\ell} \mathrm{I}_{[x_0-h,x_0+h]}(X_i)\\
&= \sum_{k=0}^{a} h^{-k}\, a_k \sum_{\ell=0}^{k} \binom{k}{\ell} (\bar{X}_0 - x_0)^{k-\ell}\\
&\quad \times \left\{ \sum_{i=1}^{n} (X_i - \bar{X}_0)^{j+\ell}\, \mathrm{I}_{[x_0-h,x_0+h]}(X_i) \right\} \qquad (13)
\end{aligned}$$

This leads to a representation of local polynomials in central (sample) moments (in braces)

$$m_j = \sum_{i=1}^{n} (X_i - \bar{X}_0)^j\, \mathrm{I}_{[x_0-h,x_0+h]}(X_i)\,. \qquad (14)$$

What remains is to find a fast and stable updating formula for m_j. For this purpose we generalized a formula for pooling estimates of variance ($j = 2$) by Chan, Golub & LeVeque (1983). Their formula is known to be fast and stable. It has been independently introduced by Spicer (1972) for the computation of central moments ($j = 1$ to 4). Suppose we have two distinct subsamples $X_{\ell 1}, \ldots, X_{\ell n_\ell}$ with means $\bar{X}_\ell$ and central moments $m_{j,\ell}$, $\ell = 1, 2$. Denote by $\bar{X}$ and m_j the mean and central moments of the union of both subsamples. Then the "add"-part of updating becomes

$$\bar{X} = \bar{X}_1 + n_2\,(\bar{X}_2 - \bar{X}_1)\,/\,(n_1 + n_2) \qquad (15)$$

and

$$\begin{aligned}
m_j &= \sum_{i=1}^{n_1} (X_{1i} - \bar{X})^j + \sum_{i=1}^{n_2} (X_{2i} - \bar{X})^j\\
&= \sum_{k=0}^{j} \binom{j}{k} (\bar{X}_1 - \bar{X})^{j-k}\, m_{k,1}\\
&\quad + \sum_{k=0}^{j} \binom{j}{k} (\bar{X}_2 - \bar{X})^{j-k}\, m_{k,2}\,. \qquad (16)
\end{aligned}$$

Note, that $m_{1,\ell} = 0$ and $m_{0,\ell} = n_\ell$. Denote

$$d = \bar{X}_1 - \bar{X}\,. \qquad (17)$$

We get $\bar{X}_2 - \bar{X} = -n_1\, d\,/\,n_2$ and for $j \geq 2$

$$\begin{aligned}
m_j &= \sum_{k=2}^{j} \binom{j}{k} d^{j-k} \left(m_{k,1} + \left(-\frac{n_1}{n_2} \right)^{j-k} m_{k,2} \right)\\
&\quad + d^j\, n_1 \left(1 - \left(-\frac{n_1}{n_2} \right)^{j-1} \right) \qquad (18)
\end{aligned}$$

A subsample is removed ("subtract"-part of updating) by

$$\bar{X}_1 = \bar{X} + n_2\,(\bar{X} - \bar{X}_2)\,/\,n_1 \qquad (19)$$

and

$$\begin{aligned}
&m_{j,1}\\
&= \sum_{i=1}^{n_1} (X_{1i} - \bar{X}_1)^j + \sum_{i=1}^{n_2} (X_{2i} - \bar{X}_1)^j - \sum_{i=1}^{n_2} (X_{2i} - \bar{X}_1)^j\\
&= \sum_{k=0}^{j} \binom{j}{k} (-d)^{j-k}\, m_k - \sum_{k=0}^{j} \binom{j}{k} (\bar{X}_2 - \bar{X}_1)^{j-k}\, m_{k,2}\,.
\end{aligned}$$

Using $\bar{X}_2 - \bar{X}_1 = -(n_1 + n_2)\, d\,/\,n_2$, as before

$$\begin{aligned}
&m_{j,1}\\
&= \sum_{k=2}^{j} \binom{j}{k} (-d)^{j-k} \left(m_k - \left(\frac{n_1 + n_2}{n_2} \right)^{j-k} m_{k,2} \right)\\
&\quad + (-d)^j\, (n_1 + n_2) \left(1 - \left(\frac{n_1 + n_2}{n_2} \right)^{j-1} \right) \qquad (20)
\end{aligned}$$

Updating the central moments (14) using these formulae results in an overall $O(n)$ algorithm.

Figure 1 shows the numerical error of the resulting fast algorithm, compared with the conventional one. Note, that the fast algorithm starts at both ends and runs to the middle of the interval. It can be seen, that centering at $x_1 = \bar{X}_0$ may have numerical advantages over centering at $x_1 = x_0$, especially at the boundary.

As can be seen, round-off errors may accumulate and restarting will be used to stabilize the updating procedure. The loss in computational speed is reduced by **"Exact binning"**: Consider a hypothetical partition of the whole sample into subsamples $X_{\ell 1}, \ldots, X_{\ell n_0}$ of length n_0 (bins). If the algorithm is restarted at x_0, say, and h is large enough, the points in the interval $[x_0 - h, x_0 + h]$ are divided into a left part with less than

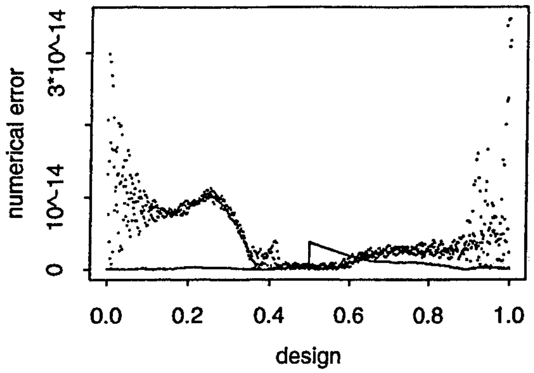


Figure 1: *Numerical error of the fast algorithm (without restarts), compared with the conventional one for the Epanechnikov weight function, $p = 1$, $n = 1000$ random uniform design points and $h = 0.25$. Solid line is fast, dots are conventional algorithm.*

n_0 observations, the central part consisting of subsamples of length n_0 (complete bins), and a remaining right part. Once a partition into bins has been chosen, the central moments of any bin are independent of the bandwidth h and the output point x_0. Hence storage of moments leads to savings in computation time: Given that central moments of such a bin have been computed, they are stored and can be used for estimation if restarting at another output point, with another bandwidth or a new (smaller) polynomial order p since they are independent of these quantities. This option of binning is particularly attractive in case of iteration as e.g. for plug–in bandwidth selection (Gasser et al., 1991) or when the same design occurs repeatedly. The following argument is helpful when choosing a bin width n_0. If the moments of the central parts are already available, the computation of m_j reduces from $O(n\,h)$ to $O(n\,h\,n_0^{-1}) + O(n_0)$ operations. Consequently n_0 should be $O((n\,h)^{1/2})$. In the usual binning only the first moments are retained which leads to an approximation error there.

The add–part (15) and (18) of the updating formula allows the construction of a moderately fast but **highly stable algorithm**: Computation of central moments of bins of length n_0 needs $O(n)$ steps. Restarting at every output point results in $O(m\,n\,h\,n_0^{-1}) + O(m\,n_0)$ operations. For $m = O(n)$, $h = O(n^{-1/5})$, and taking an optimal $n_0 = O(n^{2/5})$, we get an algorithm with $O(n^{7/5})$ operations compared to $O(n^{9/5})$ of the conventional one. The computation of central moments using exact binning is more stable than the standard two–pass algorithm, so we can expect an algorithm that is not only faster but also more stable than the conventional one. Like the conventional one this algorithm approaches $O(n)$ operations for small bandwidths $h = O(n^{-1})$.

3.4 Solution of the normal equations and automatic restart

Cholesky decomposition was used to solve the normal equations (6) for the following reasons:

- The matrices of coefficients S_n are positive definite.
- The Cholesky decomposition is fast.
- The numerical stability is scale invariant and proved to be good for the cases of interest. This fact led to the decision not to use orthogonal polynomials, which would decrease computational speed.

Also the Cholesky decomposition can be used to solve the following two numerical problems:

- control the numerical condition of the normal equations,
- control the accuracy of the updating procedure for computing the normal equations by appropriate restarting.

For this we need some theoretical analysis of Cholesky decomposition.

Cholesky decomposition: The decomposition is of the form

$$S_n = L\,D\,L'.$$

$L = ((\ell_{jk}))$ is a lower triangular matrix with diagonal elements $\ell_{jj} = 1$. $D = \operatorname{diag}(d_j)$ is the diagonal matrix of Cholesky factors. The normal equations are then solved step by step. The well known formulae for the decomposition use only the four fundamental rules of arithmetic:

$$d_j = s_{jj} - \sum_{k<j} \ell_{jk}^2\, d_k\,, \tag{21}$$

$$\ell_{jk} = \left(s_{jk} - \sum_{\ell<k} \ell_{j\ell}\,\ell_{k\ell}\,d_\ell\right) \Bigg/ d_k\,. \tag{22}$$

Cholesky factors d_j should be sufficiently away from zero compared to s_{jj} to avoid the loss of significant digits in (21). The ratios $d_j \,/\, s_{jj}$ are scale invariant. It will be shown, that they are hardly affected by the bandwidth h and by sample size n, whereas the local shape of the design density f may matter. Due to its sensitivity the last ratio $d_{p+1}\,/\,s_{p+1,p+1}$ is used to assess stability and is henceforth called "stability factor". Note, that a scale transformation, e.g. to $s_{jj} = 1$, does not improve the numerical stability of the solution.

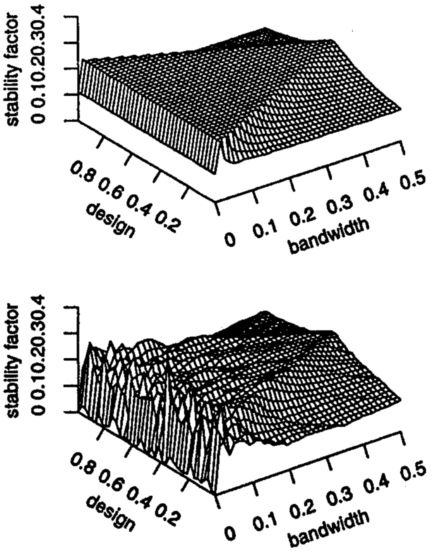


Figure 2: *Stability factor of S_n in (4) for the Epanechnikov weight function, $p = 3$ and $n = 1000$ equidistant (above) and uniformly distributed (below) design points, depending on x_0 and h.*

Under common assumptions, from (8) we get an asymptotic representation

$$
\begin{aligned}
s_{jk} & \\
&= S_{n,j+k-2} \\
&= n \int (u - x_0)^{j+k-2} K\left(\frac{u - x_0}{h}\right) f(u)\, du\, (1 + o(1)) \\
&= n\, f(x_0)\, h^{j+k-1} \int z^{j+k-2} K(z)\, dz\, (1 + o(1))\,. \quad (23)
\end{aligned}
$$

Singularity: Formula (23) leads to theoretical values of Cholesky factors d_j and the stability factor $d_{p+1} \,/\, s_{p+1,p+1}$ of S_n. For finite samples, the term $f(x_0)$ in (23) has to be replaced by a value, which only depends on the shape of the design density in $[x_0 - h, x_0 + h]$. The Cholesky factors are of order $d_j = O(n\, h^{2j-1})$ as is s_{jj}. Consequently, if the number of points in the local smoothing interval is not too small, the stability factor of S_n is near to a value, which does not depend on n and h, but only on the weight function used.

Figure 2 shows the stability factor of S_n in (4). As will be explained below, for minimal bandwidth the polynomial weight function is replaced by the uniform one. The figures show a plateau which is close to the theoretical value 0.229 even for small bandwidths. The approximation is extremely good for the fixed design. At the boundaries — increasing with h — the stability factor changes.

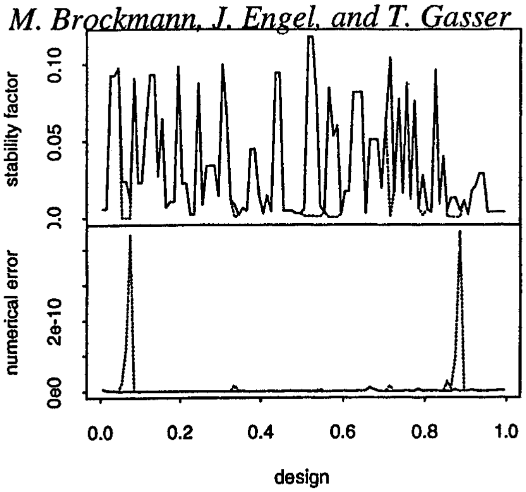


Figure 3: *Stability factor (above) of S_n in (4) for $h = 0.001$, the uniform weight function, $p = 3$ and $n = 100$ uniformly distributed design points. Solid line is stability factor for sing $= 10^{-2}$, dashes are stability factor for sing $= 10^{-30}$. Below are corresponding numerical errors.*

As to be expected a priori, and as shown by the figures, singularity is only a problem for small bandwidths. Theoretically, $p+1$ points — already required in section 2 — are sufficient to obtain a stable solution. However, in practice numerical problems may arise, basically due to two reasons. The first is that the polynomial weight function decreases the influence of points close to $x_0 \pm h$. As a first step we switch to uniform weights when there are only $p+1$ points in the interval. Then X and W are nonsingular $(p+1) \times (p+1)$ matrices, and from (6) the solution $\hat{\beta} = X^{-1}\, Y$ is independent of the weight function. Thus, the estimator is not changed, but its computation is more stable. A second reason for stability problems is, that in the random design case design points may lie close together. The independence of the stability factor of n and h gives the possibility of controlling the stability of the normal equations. S_n is defined to be singular, if

$$d_{p+1} \,/\, s_{p+1,p+1} < \text{sing} \times \text{“theoretical value”}\,,$$

where the theoretical value is derived from (23) and “sing” can be choosen by the user. However, the size of the parameter is not critical. After careful evaluation the standard value was set sing $= 0.01$. The theoretical values used depend only on p and the weight function and are given in advance. If S_n is singular, the local smoothing interval is enlarged by one point.

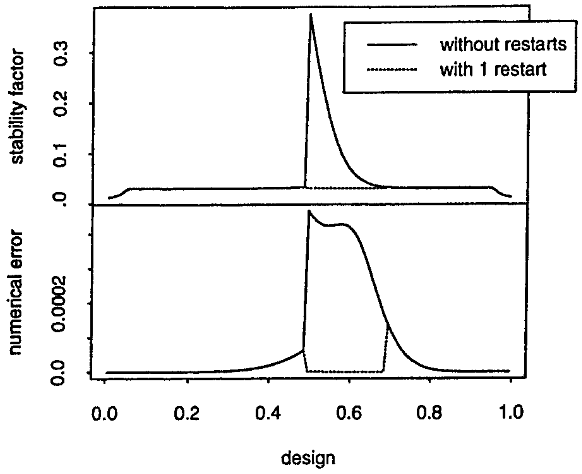


Figure 4: *Stability factor (above) of S_n in (4) for the Epanechnikov weight function, $p = 5$ and $n = 1000$ equidistant design points. Solid line is numerical stability factor without restart; dotted line is stability factor with restarts, graphically indistinguishable from the true stability factor. Below are numerical errors for $\nu = 4$ without (solid line) and with (dots) restart.*

Figure 3 shows this modification when applied to the stable $O(n^{7/5})$ algorithm described in section 3.3. Using sing $= 0.01$ only a few local smoothing intervals are changed, but the algorithm is much more stable.

Stability of updating: As noted above, the updating procedure for computing moments in the matrix S_n may lead to substantial round–off errors. The aim is to detect such departures and to restart the updating procedure. The computation of the stability factor of S_n uses all moments m_j in a complex manner, and hence allows the possibility of controlling numerical stability of the updating algorithm.

Figure 4 (above) shows the numerical stability factor of S_n in (4) without and with restarts. Note, that the algorithm starts at both ends and runs to the middle of the interval. Data are generated for a polynomial of order 5, so that a straight line for $\nu = 4$ is estimated. The figure illustrates, that the stability factor can serve as a device for detecting accumulation of round–off errors in S_n.

We use the stability factor at the last restart as benchmark, and update, as long as

$$\frac{1}{\text{stab}} < \frac{\text{“computed stability factor”}}{\text{“stability factor at last restart”}} < \text{stab}$$

The success of this restart rule using stab $= 0.95$ is demonstrated in figure 4 (below). Here is only 1 additional restart, but numerical stability is greatly improved.

4 Evaluation of algorithms

Two aims are pursued in this section:

- to check and compare numerical stability,
- to evaluate computational speed.

The scope of the evaluation is as follows:

- The range of bandwidths goes from the minimal to the maximal one.
- Interest is focussed on derivatives of order $\nu = 0, 1, 2$, while $\nu = 3, 4$ are of interest to estimate smooth functionals of $r^{(\nu)}$.
- Polynomial orders $p = \nu + 1$ are of prime interest and $p = \nu + 3$ is still of sufficient interest to warrant full evaluation. Higher order polynomials around $p = 10$ illustrate the range of applicability.

4.1 Realization of algorithms

The following three algorithms are considered:

conventional: the conventional $O(n^{9/5})$ algorithm based on (11). In fact the conventional algorithm should use (8), but for polynomial weight functions (11) is only a slight modification.

fast: the fast $O(n)$ algorithm derived in section 3.3, based on updating normal equations, exact binning, centering while moving, controlling ill–conditioned situations for small bandwidths and data–tuned restarting the updating procedure.

stable: the superstable $O(n^{7/5})$ algorithm as "fast", but restarting at every output point (no updating).

The algorithms were realized in Fortran 77 with double precision on a Sun IPX–workstation. They have additional common features:

- To reduce numerical boundary problems, the algorithms start at both ends and run to the middle of the estimation interval.
- The algorithms use Cholesky decomposition with parameters sing and stab described in section 3.4.
- When solving the normal equations, the coefficient matrix S_n is assumed to be nonsingular. In theory this is fulfilled, if the number of observations in the local smoothing interval $[x_0 - h, x_0 + h]$ is at least

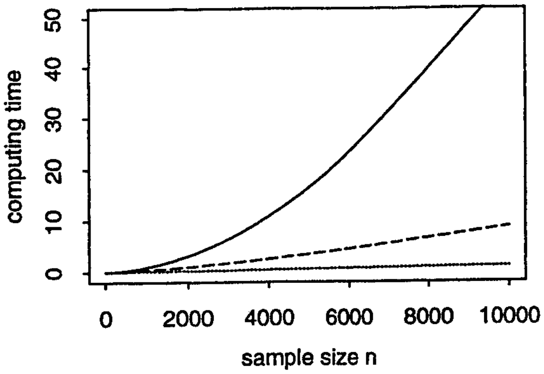


Figure 5: *Elapsed time (in seconds) of different algorithms for $\nu = 0$ and $p = 1$ depending on sample size n. Solid line is conventional, dashes are stable, and dots are fast algorithm.*

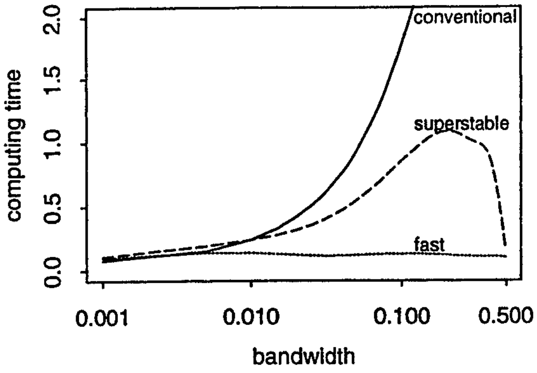


Figure 6: *Elapsed time (in seconds) of different algorithms for $\nu = 0$ and $p = 1$ depending on bandwidth (on logarithmic scale). Solid line is conventional, dashes are stable, and dots are fast algorithm.*

$p+1$. Consequently, in case of a numerically singular matrix (see section 3.4), the local bandwidth is increased.

- If the number of observations in the local smoothing interval is minimal, uniform weights are used. If this number is $p+1$, this gives the same estimator as polynomial weights. However, the unweighted estimator is numerically more stable.
- Updating saves computing time but possibly costs in numerical stability. We should restart if the situation is extremely instable or if an update does not save time. Hence a restart is forced if the number of observations is minimal, or if an update would remove more than one third of the observations used.

4.2 The design of the case study

The designs considered were fixed and random on $[0,1]$ with uniform ($f(x) = 1$), linear ($f(x) = 2x$) and truncated normal ($f(x) = \varphi(2x-1)/(2\Phi(1)-1)$) densities. The number of observations runs from $n = 10$ to 10000, focussing evaluations on $n = 1000$. Regression functions are polynomials, thereby avoiding problems with bias. Exact observations and observations with normal errors were used. The Epanechnikov weight function was chosen because of its optimality.

4.3 Computational speed

Figure 5 compares elapsed time of algorithms as a function of sample size n for random uniform design on $[0,1]$, equidistant output grid with $m = n$ points, and bandwidths $h(n) = 0.2\,n^{-1/5}$. The stable and fast algorithms used bins containing about $(2\,n\,h)^{1/2}$ observations. The fast algorithm is to a good approximation $O(n)$. From $n = 1000$ to 10000 elapsed time increased by a factor 14, slightly more than the factor 10 ideally expected. These results were confirmed for other situations.

A further point of interest is computational speed with respect to bandwidth. For fixed sample size, elapsed time of the conventional algorithm is about proportional to h. The speed of fast algorithms is expected to be approximately independent of h.

Figure 6 illustrates how elapsed time depends on h for equidistant design and output grid on $[0,1]$ with $m = n = 1000$ points. The stable and fast algorithms used bins of same length as in figure 5, i.e. they contained 10 observations. In fact, elapsed time of the fast algorithm is almost constant. For graphical reasons the time axis was cut. The conventional algorithm needed 6.9 seconds for $h = 0.5$, compared with 0.1 seconds for fast and superstable algorithms.

The elapsed time of the fast algorithm was compared with that of the fast Fourier transform (Rabiner & Gold, 1975, p. 367). Evidently, the FFT is in general not applicable to estimating the regression function r (or its derivatives) in model (1), due to inherent restrictions with respect to design, boundary problems etc. Due to its well-known good performance in terms of speed it can be taken as a benchmark in this respect. In the case whith $n = 2^k$ equidistant design points, $\nu = 0$ and $m = n$, which is ideal for the FFT, our fast algorithm needed only 70 % more time.

The attractive computational efficiency of updating algorithms has also been confirmed by Fan & Marron (1993) in a comparison with existing fast algorithms.

Table 1: *Maximal relative numerical errors rdist of algorithms over h for exact data, $m = n = 1000$ and $p = \nu + 1$, using sing $= 10^{-2}$ and stab = 0.99.*

		$\max_h \mathrm{rdist}(h)$		
design	ν	convent	stable	fast
fixed uniform	0	0.19E-13	0.19E-14	0.27E-11
	1	0.24E-11	0.11E-11	0.18E-07
	2	0.23E-08	0.12E-09	0.28E-04
	3	0.23E-05	0.52E-06	0.23E-02
	4	0.80E-02	0.23E-04	0.32
random uniform	0	0.19E-12	0.33E-14	0.30E-13
	1	0.55E-10	0.11E-09	0.11E-09
	2	0.57E-06	0.13E-07	0.13E-07
	3	0.57E-02	0.15E-05	0.15E-05
	4	0.31	0.71E-03	0.71E-03
fixed linear	0	0.55E-13	0.43E-14	0.17E-10
	1	0.13E-11	0.57E-12	0.71E-08
	2	0.33E-08	0.61E-09	0.28E-08
	3	0.10	0.44E-06	0.80E-06
	4	1.9	0.14E-03	0.62E-03
random linear	0	0.34E-13	0.13E-13	0.33E-13
	1	0.40E-10	0.63E-10	0.63E-10
	2	0.63E-01	0.83E-08	0.83E-08
	3	0.10	0.17E-05	0.17E-05
	4	2.0	0.27E-03	0.27E-03
fixed normal	0	0.41E-13	0.27E-14	0.32E-11
	1	0.18E-11	0.52E-12	0.31E-10
	2	0.29E-08	0.16E-09	0.69E-08
	3	0.27E-05	0.21E-06	0.53E-06
	4	0.97E-02	0.16E-04	0.19E-04
random normal	0	0.13E-12	0.33E-14	0.14E-13
	1	0.71E-10	0.10E-09	0.10E-09
	2	0.22E-01	0.32E-07	0.32E-07
	3	0.31E-01	0.46E-05	0.46E-05
	4	2.0	0.41E-03	0.41E-03

Table 2: *Maximal mean numerical errors mdist of algorithms over h for exact data, $m = n = 1000$ and $p = \nu + 1$, using sing $= 10^{-2}$ and stab = 0.99.*

		$\max_h \mathrm{mdist}(h)$		
design	ν	convent	stable	fast
fixed uniform	0	0.14E-14	0.81E-15	0.40E-12
	1	0.26E-13	0.13E-13	0.15E-08
	2	0.80E-11	0.72E-11	0.77E-06
	3	0.65E-08	0.17E-08	0.56E-04
	4	0.23E-04	0.26E-06	0.53E-02
random uniform	0	0.19E-13	0.13E-14	0.39E-14
	1	0.75E-12	0.97E-12	0.97E-12
	2	0.14E-08	0.14E-09	0.14E-09
	3	0.94E-05	0.28E-07	0.28E-07
	4	0.39E-03	0.42E-05	0.42E-05
fixed linear	0	0.37E-14	0.91E-15	0.20E-11
	1	0.52E-13	0.36E-13	0.27E-09
	2	0.35E-10	0.27E-10	0.44E-10
	3	0.79E-03	0.11E-07	0.11E-07
	4	0.33E-02	0.17E-05	0.17E-05
random linear	0	0.31E-14	0.80E-15	0.44E-14
	1	0.64E-12	0.75E-12	0.76E-12
	2	0.27E-03	0.16E-09	0.16E-09
	3	0.13E-02	0.29E-07	0.29E-07
	4	0.11E-01	0.33E-05	0.33E-05
fixed normal	0	0.26E-14	0.76E-15	0.86E-12
	1	0.39E-13	0.39E-13	0.61E-11
	2	0.16E-10	0.95E-11	0.10E-08
	3	0.75E-08	0.31E-08	0.31E-08
	4	0.24E-04	0.37E-06	0.37E-06
random normal	0	0.31E-14	0.11E-14	0.19E-14
	1	0.67E-12	0.11E-11	0.12E-11
	2	0.53E-04	0.15E-09	0.15E-09
	3	0.59E-04	0.27E-07	0.27E-07
	4	0.21E-02	0.40E-05	0.40E-05

4.4 Numerical stability

To check numerical stability the relative distance in sup-norm is used

$$\mathrm{rdist} = \frac{\max\limits_j \mid \hat{r}^{(\nu)}(x_j) - \tilde{r}^{(\nu)}(x_j) \mid}{\frac{1}{m} \sum\limits_{j=1}^{m} \mid \hat{r}^{(\nu)}(x_j) - \overline{\hat{r}^{(\nu)}} \mid}, \tag{24}$$

where $\hat{r}^{(\nu)}(x)$ denotes the "true" estimate, $\tilde{r}^{(\nu)}(x)$ is the result of an algorithm, and $\overline{\hat{r}^{(\nu)}}$ is the mean of true estimates. Also the following mean distance

$$\mathrm{mdist} = \frac{\sum\limits_{j=1}^{m} \mid \hat{r}^{(\nu)}(x_j) - \tilde{r}^{(\nu)}(x_j) \mid}{\sum\limits_{j=1}^{m} \mid \hat{r}^{(\nu)}(x_j) - \overline{\hat{r}^{(\nu)}} \mid} \tag{25}$$

is used. The weaker criterion "mdist" may be relevant in those cases where only a smooth functional of $r^{(\nu)}$ is of interest, as is often the case for $\nu = 3, 4$.

When inspecting stability across many situations, problems can arise typically for small bandwidths. This result should be kept in mind when judging tables 1 and 2, which give maximum numerical error across bandwidth h for supremum and for mean distance.

Table 1 shows maximal relative numerical error rdist of algorithms over $h = 0.001$, 0.005, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 for $n = 1000$ design points and $m = 1000$ equidistant output grid points. The regression functions are polynomials of order $p = \nu + 1$. The data are exact without random errors. The function $r^{(\nu)}$ to be estimated always is the straight line from -1 to 1.

Table 2 gives maximal mean numerical error mdist over h in the same situation. The numerical accuracy is good to very good for ν ranging from 0 to 3. For $\nu = 4$

the fast algorithm may break down in terms of maximal numerical error rdist, but still is useful in terms of mdist. This shows that there are only isolated problems with numerical accuracy and this has been confirmed graphically.

For $p = \nu + 3$ the precision of the superstable and fast algorithms is reduced by a factor of about 10. The conventional algorithm has problems at the boundaries for higher order polynomials because of the ill-conditioned normal equations there. The superstable and fast algorithms, however, even work stably in terms of rdist for $\nu = 1, \ldots, 4$, $p = 10, 11$, such that $p - \nu$ is odd. As expected, they are no longer fast then, and one might in these cases prefer the superstable algorithm from the beginning. The conclusions were confirmed by data with random noise and nonpolynomial regression functions.

4.5 Conclusions

We derived a fast algorithm, which is stable over the whole region of interest, i.e. up to polynomials of order about 10. The conventional algorithm has problems in terms of stability for very small bandwidths and at the boundary. The superstable algorithm proved to be more stable than the conventional one, and is at the same time much faster. It is attractive that the algorithms allow fitting of curves as well as derivatives, both for a global or local bandwidth choice.

References

Chan, T. F., Golub, G. H. & LeVeque, R. J. (1983). Algorithms for computing the sample variance: Analysis and recommendations. *Amer. Statist.* **37**, 242–247.

de Boor, C. (1978). *A Practical Guide to Splines.* New York: Springer.

Eddy, W. F. (1980). Discrete methods for statistical computations. *Proc. Statist. Comp. Sect. — ASA*, 27–31.

Fan, J. (1993). Local linear regression smoothers and their minimax efficiencies. *Ann. Statist.* **21**, 196–216.

Fan, J. & Marron, J. S. (1993). Fast implementations of nonparametric curve estimators. *Inst. Statist. Mimeo Series* #2093, Dept. Statist., Univ. NC, Chapel Hill.

Fan, J., Gasser, Th., Gijbels, I., Brockmann, M. & Engel, J. (1993). Local polynomial fitting, a framework for nonparametric regression. Manuskript.

Gasser, Th. & Kneip, A. (1989). *Ann. Statist.*, **17**, 532–535. Disscussion in: Buja, A., Hastie, T. & Tibshirani, R. (1989). Linear smoothers and additive models (with discussion). *Ann. Statist.* **17**, 453–555.

Gasser, Th., Kneip, A., & Köhler, W. (1991). A flexible and fast method for automatic smoothing. *J. Amer. Statist. Assoc.* **86**, 643–652.

Gasser, Th., Müller, H.-G. & Mammitzsch, V. (1985). Kernels for nonparametric curve estimation. *J. Roy. Statist. Soc.* **B47**, 238–252.

Härdle, W. & Scott, D. W. (1992). Smoothing by weighted averaging of shifted points. *Comput. Statist.* **7**, 97–128.

Hutchinson, M. F. & de Hoog, F. R. (1985). Smoothing noisy data with spline functions. *Numer. Math.* **47**, 99–106.

Rabiner, L. R. & Gold, B. (1975). *Theory and Application of Digital Signal Processing.* Englewood Cliffs: Prentice–Hall.

Ruppert, D. & Wand, M. P. (1992). Multivariate locally weighted least squares regression. Manuscript.

Scott, D. W. (1985). Average Shifted Histograms: effective nonparametric density estimators in several dimensions. *Ann. Statist.* **13**, 1024–1040.

Scott, D. W. (1986). Data analysis in three and four dimensions with nonparametric density estimation. In: *Statistical Image Processing and Graphics* (E. J. Wegman & D. J. de Priest, eds.), 291–305. New York: Dekker.

Seifert, B., Brockmann, M., Engel, J. & Gasser, Th. (1994). Fast algorithms for nonparametric curve estimation. *J. Comput. Graph. Statist.* **3**, 192–213.

Silverman, B. W. (1984). A fast and efficient cross-validation method for smoothing parameter choice in spline regression. *J. Amer. Statist. Assoc.* **79**, 584–589.

Spicer, C. C. (1972). Algorithm AS 52: Calculation of power sums of deviations about the mean. *Appl. Statist.* **21**, 226–227.

Utréras, D. F. (1980). Sur le choix du paramètre d'ajustement dans le lissage par fonctions spline. *Numer. Math.* **34**, 15–28.

Utréras, D. F. (1981). Optimal smoothing of noisy data using spline functions. *SIAM J. Sci. Statist. Comput.* **2**, 349–362.

Fast Implementation of Density-Weighted Average Derivative Estimation

Berwin A. Turlach

C.O.R.E & Institut de Statistique
34, Voie du Roman Pays
1348 Louvain-la-Neuve, Belgium

Abstract

Given random variables $X \in I\!R^d$ and Y such that $\mathrm{E}[Y|X=x]=m(x)$, the average derivative δ_0 is defined as $\delta_0 = \mathrm{E}[\nabla m(X)]$, i.e., as the expected value of the gradient of the regression function. Average derivative estimation has several applications in econometric theory (Stoker, 1992) and thus it is crucial to have a fast implementation of this estimator for practical purposes.

We present such an implementation for a variation known as *density-weighted average derivative estimation.* This algorithm is based on the ideas of binning or **W**eighted **A**veraging of **R**ounded **P**oints (WARPing). The basic idea of this method is to discretize the original data into a d-variate histogram and to replace in the non-parametric smoothing steps the actual observations by the appropriate bincenters. The non-parametric smoothing steps become thus a (multi-dimensional) convolution between the (discretized) data and the (discretized) smoothing kernel.

A Monte-Carlo study demonstrates that with this binned implementation substantial reduction in computing time can be achieved. But it will also become clear that in higher dimension the choice of **how to bin** is crucial.

1 Introduction

Average derivative estimation tries to estimate the mean slope of the conditional mean of the response variable, i.e., given a response variable Y, whose expectation is assumed to depend on a d-dimensional variable X via a smooth function m, the aim of average derivative estimation is to estimate the average slope of this function. In other words, if

$$\mathrm{E}[Y|X=x]=m(x)$$

and ∇ denotes the gradient of partial derivatives with respect to the coordinates of X, the aim is to estimate

$$\delta_0 = \mathrm{E}[\nabla m(X)] \tag{1}$$

respectively a weighted version

$$\delta_w = \mathrm{E}[\nabla m(X) w(X)] \tag{2}$$

where $w(\bullet)$ is a non-negative weight function. If we choose as weight function $w(x) \equiv f(x)$, the marginal density of X, our estimand becomes:

$$\begin{aligned}\delta &= \mathrm{E}[\nabla m(X) f(X)] \\ &= -2\mathrm{E}[Y \nabla f(X)] \end{aligned} \tag{3}$$

Where (3) follows by partial integration. The problem of estimating the *density-weighted average derivative*, as given by (3), was studied by Powell, Stock and Stoker (1989).

Average derivative estimation can be used in many econometric models (Stoker, 1992; Härdle, Hildenbrand and Jerison, 1991). As one example, we want to mention *single-index* models (also called *one-term projection pursuit* models). In these models the regression function m has the form

$$m(x) = g(x^T\beta), \tag{4}$$

where g is an unknown univariate function and β is a d-dimensional (projection) vector. Stoker (1986) gives an extensive discussion and motivation for models of the form (4). The semiparametric model (4) covers a broad range of important parametric models such as probit and logit models, censored regression, Tobit models etc.

It is easy to see, that in this case we have

$$\nabla m(x) = g'(x^T\beta)\beta$$

and thus

$$\delta_0 = \mathrm{E}[g'(X^T\beta)]\beta \quad \text{and} \quad \delta_w = \mathrm{E}[g'(X^T\beta)w(X)]\beta.$$

This means that (weighted) average derivative estimation allows us to estimate the unknown projection β up to a scale constant. This is in fact the best we can do in the semiparametric single-index model given by (4). If the pair (g, β) fulfills model (4) then for any $c \in \mathbb{R}$, $c \neq 0$, the pair $(\tilde{g}, \tilde{\beta})$ with

$$\tilde{g}(\bullet) = g(\bullet/c) \quad \text{and} \quad \tilde{\beta} = c\beta$$

does so too.

The rest of this article is structured as follows, Section 2 will describe the density-weighted average derivative estimator as proposed by Powell et al. (1989). In Section 3 we will propose how to implement this estimator using binning ideas and to achieve thus considerable run-time gains. Finally in Section 4 we will discuss some further points related to the binning method.

2 Direct implementation

2.1 Estimator for δ

To estimate the density-weighted average derivative δ, Powell et al. (1989) propose to estimate the gradient of the marginal density of the X variables nonparametrically at each observation point by, say, $\widehat{\nabla f}(x_i)$. Their estimator for δ is

$$\hat{\delta} = -\frac{2}{n}\sum_{i=1}^{n} y_i \widehat{\nabla f}(x_i) \tag{5}$$

which can be motivated as a method of moment estimator in which the unknown function ∇f is replaced by a nonparametric estimate of it.

To estimate the gradient of f nonparametrically, Powell et al. (1989) use the gradient of a multivariate kernel density estimator (Silverman, 1986; Scott 1992). Given a d-variate kernel $\mathcal{K}$ (think of $\mathcal{K}$ as a d-variate density function) and a $d \times d$ positive definite matrix H of smoothing parameters a nonparametric estimate of the marginal density f at a point $x \in \mathbb{R}^d$ would be

$$\hat{f}_H(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{\det(H)} \mathcal{K}\left(H^{-1}(x - x_i)\right). \tag{6}$$

For numerical ease, a common choice is to take $\mathcal{K}$ as a product of d univariate kernels K, and to reduce H to a diagonal matrix, so that we have only a d-dimensional vector h of smoothing parameters. Wand and Jones (1993) discuss for the two-dimensional case the implications of this simplification. With this choices (6) simplifies to

$$\hat{f}_h(x) = \frac{1}{nh_1 \dots h_d}\sum_{i=1}^{n}\prod_{k=1}^{d} K\left(\frac{x_k - x_{ik}}{h_k}\right). \tag{7}$$

where $x = (x_1, \dots, x_d)^T$ and $x_j = (x_{j1}, \dots, x_{jd})^T$.

Powell et al. (1989) do not use the nonparametric density estimator given in (7) directly, but a *leave-one-out* version of it. (For this reason the estimator $\hat{\delta}$ has a U-statistic structure and can be easily analyzed.) Thus to estimate the marginal density f at the observation x_i, they drop x_i from the sample and calculate $\hat{f}_h(x_i)$ from the remaining sample (of size $n-1$). As a further simplification they use only *one* bandwidth for all dimensions. So the estimator $\widehat{\nabla f}(x_i)$ which they use in (5) is:

$$\begin{aligned}\widehat{\nabla f}(x_i) &= \frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n} \frac{1}{h^{d+1}}\mathcal{K}'\left(\frac{x_{ik} - x_{jk}}{h}\right) \qquad (8)\\ &= \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_d}\end{pmatrix}\left(\frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n}\prod_{k=1}^{d} K_h\left(x_{ik} - x_{jk}\right)\right).\end{aligned}$$

with $K_h(u) = K(u/h)/h$.

2.2 Asymptotic properties

Powell et al. (1989) showed that under certain regularity conditions and a suitable choice for K and the rate with which h tends to zero, the estimator $\hat{\delta}$ given in (5) is consistent and has an asymptotic normal distribution. More specifically they proved that

$$\sqrt{n}\left(\hat{\delta} - \delta\right) \xrightarrow{\mathcal{L}} \mathcal{N}(0, \Sigma)$$

where

$$\begin{aligned}\Sigma &= 4\mathrm{E}[r(X,Y)r(X,Y)^T] - 4\delta\delta^T, \\ r(x,y) &= f(x)\nabla m(x) - \{y - m(x)\}\nabla f(x).\end{aligned}$$

2.3 Estimator for the variance

To estimate the asymptotic variance Σ of $\hat{\delta}$ Powell et al. (1989) propose to estimate $r(x_i, y_i)$ by:

$$\hat{r}(x_i, y_i) = -\frac{1}{n-1}\sum_{\substack{j=1\\ j\neq i}}^{n}\frac{1}{h^{d+1}}\mathcal{K}'\left(\frac{x_i - x_j}{h}\right)(y_i - y_j) \tag{9}$$

and thus Σ by:

$$\hat{\Sigma} = 4\frac{\sum_{i=1}^{n}\hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T}{n} - 4\hat{\delta}\hat{\delta}^T. \tag{10}$$

In the next section we will discuss how fast implementations for $\hat{\delta}$ and $\hat{\Sigma}$ can be obtained by using binning techniques.

3 Binned implementation

3.1 Basic idea

The basic idea of binning methods is to replace each observation of x_i by the nearest point b_z from a regular spaced grid. To fix ideas consider kernel density estimation in the one-dimensional case,

$$\hat{f}_h(x_i) = \frac{1}{n}\sum_{j=1}^{n} K_h(x_i - x_j), \tag{11}$$

and take the regular grid $\{b_z : b_z = z\Delta, z \in \mathbb{Z}\}$ where Δ is a fixed constant, the *binwidth*. Replacing now each x_i in (11) by the nearest b_z, we see that we have to evaluate the kernel K only at integer multiple of Δ/h:

$$w_l = \frac{1}{h}K\left(\frac{\Delta}{h}l\right), \quad l = -L, \ldots L \tag{12}$$

Here L is chosen such that $\Delta L/h \approx 1$ if K has compact support on $[-1,1]$ (if K is the Gaussian kernel, i.e., the kernel has no compact support, Wand (1993) recommends $\Delta L/h \approx 4$). If we denote further by n_z the number of observations x_i which have b_z as their nearest point in the grid, we see that we can approximate (11) by (let b_z be the point nearest to x_i):

$$\begin{aligned}
\hat{f}_h(x_i) &= \frac{1}{n}\sum_{j=1}^{n} K_h(x_i - x_j) \\
&\approx \frac{1}{n}\sum_{j=1}^{n} w_{z-l_j}, \quad b_{l_j} \text{ is nearest to } x_j \\
&= \frac{1}{n}\sum_{l=-L}^{L} w_{z-l} n_l.
\end{aligned}$$

The last formula is a discrete convolution between the vector of weights (the discretized kernel) and the vector of *bincounts* n_z (the discretized data).

Silverman (1982) uses a fast fourier transformation to calculate this discrete convolution. Another algorithm which does not use the fast fourier transform is given in Scott (1985) (see also Härdle and Scott, 1992; Härdle, 1991). Fan and Marron (1994) describe how to use these ideas for other nonparametric curve smoothers.

Fan and Marron (1994) also quantify the run-time gains achievable using these ideas. These run-time gains are mainly due to two facts. First we have much less kernel evaluations, in fact we have to evaluate the kernel only once on a finite grid of points. Secondly, once the data is discretized the nonparametric curve smoother is estimated at the grid points b_z and not at the original observations x_i. Usually the number of grid points at which the smoother is evaluated is (much) smaller than n. The estimate at an original observation x_i is either taken as the estimate at the nearest b_z or obtained by linear interpolation between the estimates of the two nearest grid points (Jones, 1989).

3.2 Application to $\hat{\delta}$

The ideas presented in Section 3.1 above are readily extendable to the multivariate case (Wand, 1993) and to the estimator $\hat{\delta}$.

Again we define a (multivariate) grid of equidistant points $b_z \in \mathbb{R}^d$ and replace $x_i \in \mathbb{R}^d$ by the nearest b_z. To fix ideas let $\Delta = (\Delta_1, \ldots, \Delta_d)^T$ be a fixed d-dimensional vector and define b_z by

$$b_z = z\Delta = (z_1\Delta_1, \ldots, z_d\Delta_d)^T$$

for each multi-index $z = (z_1, \ldots, z_d)^T \in \mathbb{Z}^d$. Note the pointwise multiplication of the vectors z and Δ above. In the rest of this article, if not indicated differently, we mean this kind of pointwise vector multiplication rather then the standard matrix multiplication when we multiply two vectors.

For each $z \in \mathbb{Z}^d$, let again n_z denote the number of observed x_i for which b_z is the nearest grid point. For a binned implementation of the estimator $\widehat{\nabla f}$ we also need to discretize the derivative of the kernel K:

$$\tilde{w}_{lj} = \frac{1}{h^2}K'\left(\frac{\Delta_j}{h}l\right), \quad \begin{array}{l} l = -L_j, \ldots, L_j \\ j = 1, \ldots, d \end{array} \tag{13}$$

and define w_{lj} analogous to (12) by replacing Δ by Δ_j. If we define now for each multi-index $l = (l_1, \ldots, l_d)^T \in \mathbb{Z}^d$ the corresponding weight $w'_l \in \mathbb{R}^d$ by:

$$w'_l = \begin{pmatrix} \tilde{w}_{l_1 1} w_{l_2 2} \cdots w_{l_d d} \\ w_{l_1 1} \tilde{w}_{l_2 2} \cdots w_{l_d d} \\ \vdots \\ w_{l_1 1} w_{l_2 2} \cdots \tilde{w}_{l_d d} \end{pmatrix}$$

we see that analogous to the example in Section 3.1 a binned version of the estimator $\widehat{\nabla f}$ is:

$$\widehat{\nabla f}(b_z) = \frac{1}{n-1}\sum_{l=-L}^{L} w'_{z-l} n_l. \tag{14}$$

Note that the sum in (14) is actually a sum over d indices $l_1, \ldots, l_d$, each l_j taking values from $-L_j$ to L_j, $j = 1, \ldots, d$. Also, the multi-index $z - l$ in (14) is $z - l = (z_1 - l_1, \ldots, z_d - l_d)^T$.

Thus a binned version of the density-weighted average derivative δ is:

$$\hat{\delta} = -\frac{2}{n}\sum_{z \in \mathbb{Z}^d} n_z \bar{y}_z \widehat{\nabla f}(b_z) \tag{15}$$

where $\bar{y}_z$ is the average over all observation y_i such that b_z is the nearest grid point to the corresponding x_i. Note that the summation in (15) is actually only over all $z \in \mathbb{Z}^d$ such that $n_z \neq 0$ and *is not* an infinite sum. Furthermore, if we compare (5) with (15) we see that the only approximation error we do is due to replacing $\widehat{\nabla f}(x_i)$ by $\widehat{\nabla f}(b_z)$. With respect to the y we "keep the full resolution".

3.3 Application to $\hat{\Sigma}$

In this section we will discuss the implementation of a binned estimator for the asymptotic variance Σ given in Section 2.2. A naive way of implementing such an estimator would be to plug into (10) a binned estimate, say, $\hat{r}(b_z)$ for $\hat{r}(x_i, y_i)$, given in (9), to obtain:

$$\hat{\Sigma} = 4\frac{\sum_{z\in\mathbb{Z}^d} \hat{r}(b_z)\hat{r}(b_z)^T}{n} - 4\hat{\delta}\hat{\delta}^T \tag{16}$$

with $\hat{\delta}$ from (15). The binned estimate $\hat{r}(b_z)$ is easily derived in the same way as demonstrated in Section 3.1. Let b_z be the grid point nearest to x_i, then we have:

$$\begin{aligned}
&\hat{r}(x_i, y_i) = \\
&\quad = -\frac{1}{n-1}\sum_{j=1}^{n} \frac{1}{h^{d+1}}\mathcal{K}'\left(\frac{x_i - x_j}{h}\right)(y_i - y_j) \\
&\quad \approx -\frac{1}{n-1}\sum_{j=1}^{n} w'_{z-l_j}(y_i - y_j), b_{l_j} \text{ is nearest to } x_j \\
&\quad = -\frac{1}{n-1}\sum_{l=-L}^{L} w'_{z-l} n_l (y_i - \bar{y}_l) = \hat{r}(b_z, y_i) \\
&\quad \approx -\frac{1}{n-1}\sum_{l=-L}^{L} w'_{z-l} n_l (\bar{y}_z - \bar{y}_l) = \hat{r}(b_z)
\end{aligned}$$

Note that the only approximation error in $\hat{r}(b_z, y_i)$ is due to replacing the x_i by the grid point b_z. Thus for $\hat{r}(b_z, y_i)$ we have still the full resolution in the y-direction. Only if we go to $\hat{r}(b_z)$ we make an approximation error in that direction too. The motivation for this approximation is, that if several x_i exist which have b_z as nearest grid point then we should average over the corresponding $\hat{r}(b_z, y_i)$ to get a unique estimate $\hat{r}(b_z)$ at b_z.

However, the binned implementation which we get if we insert $\hat{r}(b_z)$ in (16) *does not* work. The reason for this is explained and graphically illustrated in Proença and Turlach (1994). On one side we make an approximation error in the y-direction by going from $\hat{r}(b_z, y_i)$ to $\hat{r}(b_z)$. On the other side we want to approximate $\hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T$ which involves a squared term in y. Thus we have to take into account what Proença and Turlach (1994) call the *within-bin-variability* of y. This means that we can not find a binned estimator for $\hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T$ by finding one just for $\hat{r}(x_i, y_i)$, but that we really have to consider this product directly. Hence a "correct" binned estimator can be found by observing that:

$$\begin{aligned}
&\hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T \approx \\
&\quad \approx \hat{r}(b_z, y_i)\hat{r}(b_z, y_i) \\
&\quad = \left(\frac{1}{n-1}\right)^2 \sum_{l=-L}^{L}\sum_{l'=-L}^{L} w'_{z-l} w'^T_{z-l'} \times \\
&\qquad\qquad n_l(y_i - \bar{y}_l) n_{l'}(y_i - \bar{y}_{l'}) \\
&\quad = \left(\frac{1}{n-1}\right)^2 \sum\sum_{l,l'=-L}^{L} \Big\{ w'_{z-l} w'^T_{z-l'} \times \\
&\qquad\qquad n_l(y_i - \bar{y}_z + \bar{y}_z - \bar{y}_l) n_{l'}(y_i - \bar{y}_z + \bar{y}_z - \bar{y}_{l'}) \Big\} \\
&\quad = \hat{r}(b_z)\hat{r}(b_z)^T + \\
&\qquad + \left(\frac{1}{n-1}\right)^2 \sum\sum_{l,l'=-L}^{L} \Big\{ w'_{z-l} w'^T_{z-l'} \times \\
&\qquad\qquad n_l n_{l'}(y_i - \bar{y}_z)(2\bar{y}_z - y_l - y_{l'}) \Big\} \\
&\qquad + \left(\frac{1}{n-1}\right)^2 \sum\sum_{l,l'=-L}^{L} w'_{z-l} w'^T_{z-l'} n_l n_{l'} (y_i - \bar{y}_z)^2
\end{aligned}$$

And thus the sum $\sum_{i=1}^{n} \hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T$ can be approximated as:

$$\begin{aligned}
&\sum_{i=1}^{n} \hat{r}(x_i, y_i)\hat{r}(x_i, y_i)^T \approx \\
&\quad \approx \sum_{z\in\mathbb{Z}^d}\sum_{i=1}^{n} \hat{r}(b_z, y_i)\hat{r}(b_z, y_i)^T \\
&\quad = \sum_{z\in\mathbb{Z}^d} \Big\{ \hat{r}(b_z)\hat{r}(b_z)^T + \\
&\qquad \left(\frac{1}{n-1}\right)^2 \sum\sum_{l,l'=-L}^{L} w_{z-l} w'^T_{z-l'} n_l n_{l'} n_z (\overline{y^2_z} - \bar{y}^2_z) \Big\} \\
&\quad = \sum_{z\in\mathbb{Z}^d} \Big\{ \hat{r}(b_z)\hat{r}(b_z)^T + n_z(\overline{y^2_z} - \bar{y}^2_z)\widehat{\nabla f}(b_z)\widehat{\nabla f}(b_z)^T \Big\} \\
&\quad = \widehat{rr^T}
\end{aligned}$$

Note that because of the summation over i the term which includes $(y_i - \bar{y}_z)(2\bar{y}_z - \bar{y}_l - \bar{y}_{l'})$ drops out, i.e., the sum is zero. Also, $\bar{y}^2_z$ denotes the square of $\bar{y}_z$ and $\overline{y^2_z}$ denotes the mean of all y^2_i such that x_i has b_z as nearest grid point. This term, namely $n_z(\overline{y^2_z} - \bar{y}^2_z)$, measures the variability of Y around the grid point b_z. This term is obtained by expanding $(y_i - \bar{y}_z)^2$ and summing over i. Note that if we choose Δ so small, that each grid point b_z has at most one observation x_i for which it is the nearest point then all of these within-bin-variability terms vanish and the binned estimator given in (16) would be correct.

However, in general we have to take these terms into account. Thus a "correct" binned estimator for the variance matrix is given by

$$\hat{\Sigma} = 4\frac{\widehat{rr^T}}{n} - 4\hat{\delta}\hat{\delta}^T$$

with $\hat{\delta}$ from (15).

4 Closing remarks

In the previous section we demonstrated how the simple and intuitive basic binning idea can be applied to the density-weighted average derivative estimator $\hat{\delta}$ and the estimator of the asymptotic covariance matrix $\hat{\Sigma}$. Some questions still remain which we would like to address here.

From (14) we see that $\widehat{\nabla f}(b_z)$ is a discrete convolution, the same is true for $\hat{r}(b_z)$ and $\widehat{rr^T}$. How should we calculate this discrete convolution? As mentioned above Silverman (1982) and Wand (1993) use a fast fourier transformation. However, this method is inappropriate in our situation since we are only interested to calculate these estimates at the points b_z which have some observation close enough to them, i.e., for which $n_z \neq 0$. But a fast fourier transformation method would calculated these estimates at *all* grid points b_z. Just imagine the case where we have a two-dimensional X-variable and we choose our grid such that we have 100 different grid points in each dimension. The complete grid will have 10.000 points b_z. In this case a fast fourier transform method would calculate $\widehat{\nabla f}(b_z), \ldots$ at all these grid points. Clearly this involves many unnecessary calculations if the sample size is not too big.

The fast fourier transform approach is feasible if we need estimates at all grid points for example if we want to make a plot. But it is also not clear if the fast fourier transform is the fastest method in such a case. Fan and Marron (1994) find that this approach is not the fastest for the one-dimensional case whereas Wand (1993) favors the fast fourier transform in the two-dimensional case. Scott (1992) describes alternative algorithms which do not use a fast fourier transform. These algorithm step through all grid points b_z with $n_z \neq 0$ and just do the necessary calculations at these points *and* in the neighborhood of b_z (as defined by the L_j), i.e., also these algorithms calculate the estimates on the whole grid. For the discrete convolution necessary here we recommend to use specialized versions of the algorithms of Scott (1994) which step through all grid points b_z with $n_z \neq 0$ and do the necessary calculations *only* at these points.

Closely related with the question "How to perform the discrete convolution?" is the question "How shall one discretize the data?". Until now we always used a kind of "histogram" binning in which n_z was integer and each observation was shifted to (replaced by) the nearest grid point b_z. For the one-dimensional density estimation Jones and Lotwick (1984) proposed an alternative called "linear" binning. In this variation the n_z are no longer integer and each observation is distributed onto the *two* nearest grid points. Hall and Wand (1993) propose further variations for the binning procedure and quantify the error which is introduced by using binning techniques (see also González-Manteiga, Sánches-Sellero and Wand, 1994).

But the use of such techniques in a higher-dimensional setting is problematic. A binning technique like "linear" binning which distributes each observation in one-dimension on two grid points, will distribute each observation in d-dimension onto 2^d grid points. This could have the effect that we have more grid points b_z with $n_z \neq 0$ than observations! Take for example a two-dimensional standard normal variable and use linear binning with a grid where $\Delta = (0.03, 0.03)^T$. If the sample size is $n = 250$ we have on the average 950 grid points b_z at which $n_z \neq 0$. The result of this is that, even if we use the algorithms described above for the discrete convolution, the binned implementation using "linear" binning is *slower* than the direct implementation.

This was verified in a Monte-Carlo study with a bivariate X-variable (and Y generated according to a linear model and a probit model). Using the adapted algorithms from Scott (1992) for the discrete convolution and "linear" binning hardly no run-time gains were observed and for a grid with small Δ the direct implementation was even faster. If "histogram" binning was used, however, we observed run-time gains of a factor 10 over the direct implementation.

Thus we recommend to use "histogram" binning and the (adapted) algorithms of Scott (1993) for functional estimation in higher dimensions.

Acknowledgments

I would like to thank Steve Marron and Matt Wand for the discussions we had about binning techniques. Likewise I would like to thank Tom Stoker and Jim Powell for discussions about (weighted) average derivative estimation. This research was supported by the EC grant B/SPES 915011.

References

J. Fan and J. S. Marron. Fast implementations of nonparametric curve estimators. *Journal of Computational and Graphical Statistics*, 3(1):35–56, 1994.

W. González-Manteiga, C. Sánchez-Sellero, and M. P. Wand. Accuracy of binned kernel functional estimators. unpublished manuscript, 1994.

P. Hall and M. P. Wand. On the accuracy of binned kernel density estimators. Working Paper 93–003, The University of New South Wales, Australian Graduate School of Management, PO Box 1, Kensington NSW 2033, Australia, 1993.

W. Härdle. *Smoothing Techniques, With Implementations in S.* Springer, New York, 1991.

W. Härdle, W. Hildenbrand, and M. Jerison. Empirical evidence on the law of demand. *Econometrica*, 59(6):1525–1549, 1991.

W. Härdle and D. W. Scott. Smoothing by weighted averaging of rounded points. *Computational Statistics*, 7:97–128, 1992.

M. C. Jones. Discretized and interpolated kernel density estimates. *Journal of the American Statistical Association*, 84(407):733–741, 1989.

M. C. Jones and H. W. Lotwick. A remark on algorithm AS176: Kernel density estimation using the fast fourier transform (Remark ASR50). *Applied Statistics*, 33:120–122, 1984.

J. L. Powell, J. H. Stock, and T. M. Stoker. Semiparametric estimation of index coefficients. *Econometrica*, 57(6):1403–1430, 1989.

I. M. Proença and B. A. Turlach. Fast implementation of bandwidth selectors. unpublished manuscript, 1994.

D. W. Scott. Averaged shifted histograms: Effective nonparametric density estimators in several dimensions. *Annals of Statistics*, 13(3):1024–1040, 1985.

D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, New York, Chichester, 1992.

B. W. Silverman. Kernel density estimation using the fast fourier transform. Statistical algorithm AS 176. *Applied Statistics*, 31:93–97, 1982.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*, volume 26 of *Monographs on Statistics and Applied Probability*. Chapman and Hall, London, 1986.

T. M. Stoker. Consistent estimation of scaled coefficients. *Econometrica*, 54(6):1461–1481, 1986.

T. M. Stoker. *Lectures on Semiparametric Econometrics.* Center for Operation Research and Econometrics, Université Catholique de Louvain, Voie du Roman Pays 34, 1348 Louvain-la-Neuve, Belgium, 1992.

M. P. Wand. Fast computation of multivariate kernel estimators. Working Paper 93–007, The University of New South Wales, Australian Graduate School of Management, PO Box 1, Kensington NSW 2033, Australia, 1993.

M. P. Wand and M. C. Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88:520–529, 1993.

Experiences With Derivative-Free REML

L. Dale Van Vleck
ARS-USDA, U.S. Meat Animal Research Center and
University of Nebraska, Lincoln, NE

Abstract

A set of Fortran programs has been developed to obtain (co)variance estimates for multiple trait genetic analyses with different models for each trait using the sparse matrix package SPARSPAK, and a derivative-free algorithm to obtain REML estimates (MTDFREML). A typical analysis would include birth weight of all animals, weaning weight and yearling weight on those surviving. The model would include direct genetic and correlated maternal genetic effects for each animal and uncorrelated maternal environmental effects (a total of 33 (co)variance components) as well as other fixed or random effects associated with the traits. The simplex algorithm is used to search for components to minimize -2 log likelihood = FVALUE. The FVALUE for equations of order 60,000 or more can be evaluated on personal computers for each of the potentially thousands of rounds needed to obtain REML estimates. Efficiency depends on density of the mixed model equations. Nongenetic models are usually much more sparse than genetic models that incorporate numerator relationships among the animals. Scaling of variables is sometimes a problem due to rounding in calculation of FVALUE; e.g., multiplying categorical variables by 100 led to successful convergence. The search algorithm is stopped when variance for FVALUEs in the Simplex is from 10^{-4} to 10^{-8}, often at a local minimum. With multiple trait analyses, several restarts may be needed to find the global maximum. An evolving strategy is:

1. begin with only variances included to minimum local convergence.
2. restart with covariances included to minimum local convergence until FVALUE change is no more than a unit.
3. restart with maximum local convergence (10^{-6} to 10^{-8}) until FVALUE change is only at second or third decimal when global maximum is declared.

Successful analyses with MTDFREML require "art" as well as "science".

Introduction

Restricted maximum likelihood (Patterson and Thompson, 1972) has become the preferred method of animal breeders to estimate (co)variance matrices among and within traits described by mixed linear models. The traditional algorithms make use of identities based on Henderson's (e.g., 1963, 1984) mixed model equations which have computational advantages including being based on a simple modification of least squares equations. Algorithms based on derivatives of the multivariate normal likelihood given the data have been limited in scope by requiring inverse elements of the coefficient matrix of the mixed model equations. For practical purposes, that has meant mixed model equations with order in the range of 1000-5000.

Derivative-free algorithms that take advantage of the sparsity of the coefficient matrix have greatly expanded the number of equations that can be managed to the order of 50,000 to 150,000. The purpose of this note is to outline briefly the science of DFREML and then to discuss some aspects of the "art" of DFREML as the numerical properties are not well understood, at least to most animal breeders.

The Science of DFREML

The original algorithm for DFREML as developed in animal breeding traces to several sources including the realization that Gaussian elimination of augmented least squares (although in this case, mixed model) equations can be used to obtain the two computing intensive parts of the log likelihood (Smith and Graser, 1986; Graser, Smith and Tier, 1987) as the keynote speaker for this conference described (Stewart, 1994). The other two developments were Hendersons' mixed model equations (e.g., 1963) and the discovery that the log of the likelihood can be written in terms of four components of the mixed model equations (Harville, 1977; Searle, 1979).

The general linear model in typical animal breeding notation is:

$$y = X\beta + Zu + e$$

$$E[y] = X\beta$$

$$V\begin{pmatrix} u \\ e \end{pmatrix} = \begin{pmatrix} G & 0 \\ 0 & R \end{pmatrix}, \text{ and}$$

$$V(y) = V = ZGZ' + R$$

where **y** is the vector of observations; **ß** is the vector of fixed effects with association matrix, **X**; **u** is the vector of random effects with association matrix, **Z**; and (co)variance matrix, **G**; and **e** is the vector of residuals associated with the observations with (co)variance matrix, **R**.

Henderson (e.g., 1984) showed that solutions to mixed model equations provide best linear unbiased estimators of estimable functions of fixed effects and best linear unbiased predictors of realized values of random effects.

Henderson's mixed model equations (MME) are:

$$\begin{pmatrix} X'R^{-1}X & X'R^{-1}Z \\ Z'R^{-1}X & Z'R^{-1}Z + G^{-1} \end{pmatrix}\begin{pmatrix} \hat{\beta} \\ \hat{u} \end{pmatrix} = \begin{pmatrix} X'R^{-1}y \\ Z'R^{-1}y \end{pmatrix}$$

In simpler notation: C s = r.

Note that except for the usual zero covariance between the **u** and **e** vectors, the mixed model equations are completely general and can encompass multiple traits, missing observations on some traits of some animals, different models for different traits and, for animal breeders, relationships among animals due to genes in common, **A**, and genetic covariances among traits, G_0.

Typical random factors in animal breeding models include animal's direct genetic value, mother's maternal genetic value (with genetic covariance between direct and maternal genetic values), animal permanent environmental effects when animals have repeated records, and maternal permanent environmental effects when mothers have more than one progeny with records. Other genetic models used by animal breeders may include instead of animal effects, sire transmitting ability (1/2 direct genetic value of sire), maternal grandsire effect and dam permanent environmental effect. Other variations are also used.

The large number of variances and covariances to estimate from multiple trait models can be illustrated for traits with a direct and maternal genetic value (with covariance) and two other random factors such as dam permanent environmental and a litter effect in addition to residual effects. A single trait analysis will involve five variances and one covariance. A two-trait analysis will involve those six elements twice plus seven other covariances. A three-trait analyses would have 6 + 6 + 6 + 7 + 7 + 7 = 39 variance and covariance components to estimate.

Harville (1977) and Searle (1979) showed that the multivariate normal likelihood given the data is:

$$\Lambda = -.5[\text{constant} + \log|R| + \log|G| + \log|C| + y'Py] \text{ where}$$

$$C = \text{coefficient matrix for MME and}$$

$$P = V^{-1} - V^{-1}X(X'V^{-1}X)^{-1}X'V^{-1}$$

Note that C and P depend on R and G as well as on X and Z.

Derivative-Free Algorithms

Derivative-free algorithms for REML are based on searching for the combination of individual variances and covariances associated with R and G that will maximize Λ or, more usually, will minimize, FVALUE = -2Λ. The original algorithm of Smith and Graser (1986) and that used in the single trait program of Meyer (1988) which popularized use of DFREML was based on sparse matrix Gaussian elimination of C augmented with **r** with the total sum of squares in the corresponding diagonal. Gaussian elimination automatically produced a known multiple of y'Py and log |C|, the difficult-to-compute terms in Λ. The simplex algorithm (Nelder and Mead, 1965) is the usual choice to search for (co)variances to minimize -2Λ.

Boldman and Van Vleck (1991) used subroutines in SPARSPAK (George, et al., 1980; Chu, et al., 1984) to decrease the time to calculate -2Λ by factors of 100 to 600 from the times required by the original algorithm of Meyer (1988). SPARSPAK is based on Choleski factorization rather than Gaussian elimination and provides a more general form for calculation of y'Py as well as log |C|. Both the Gaussian and Choleski based algorithms lead to general programs which are not model dependent, whereas derivative based algorithms are more difficult to generalize because of the requirements to calculate a quadratic in **y** for each (co)variance component and to calculate the expectation of the quadratic which is a function of corresponding elements of the inverse of the coefficient matrix.

Some general observations are that derivative based algorithms are slow to converge, that single trait DFREML converges quickly but that multi-trait analyses may converge slowly with DFREML. Many restarts may be needed if covariances are estimated (Press, et al., 1989; Groeneveld and Kovak, 1990; Boldman and Van Vleck, 1990).

The Choleski based algorithm used for the MTDFREML package (Boldman, et al., 1993) consists of two basic steps:

1. a method (the simplex algorithm) to search for parameter estimates to minimize -2Λ and
2. formation and solution of MME for parameter estimates chosen by the simplex algorithm by use of SPARSPAK subroutines to take advantage of the usual sparsity of the mixed model equations.

Their package also includes a program to calculate the inverse of the relationship matrix among the animals to be used in forming the mixed model equations (Quaas, 1976) and a preparation program which recodes animal identification and fixed effect levels into equation numbers.

Calculation of -2Λ

If y_i is the vector of observations on traits measured on animal i, then the residual covariance matrix for animal i is R_i. For the usual assumption that residuals from one animal to another are uncorrelated, then $\log|R| = \Sigma \log|R_i|$ where each R_i is dependent on the number of traits measured on animal i. All eigenvalues of R_0, the maximum order of any R_i, must be positive. Thus, one way to calculate $\log|R|$ is to calculate the sum of logarithms of eigenvalues for each type of R_i and multiply by the number of each type of R_i and then sum over all types of R_i. The $\log|G|$ can be calculated similarly and even more easily (e.g., Meyer, 1989, 1991). For example, if:

$$G = \begin{pmatrix} A\otimes G_o & 0 & 0 \\ 0 & I_1\otimes C_{11} & 0 \\ & \ddots & \vdots \\ 0 & 0 & I_L\otimes C_{LL} \end{pmatrix}$$

then

$$\log|G| = t\log|A| + q\log|G_o| + n_1\log|C_{11}| + \cdots + n_L\log|C_{LL}|$$

where **t** is the order of G_0 which is the genetic covariance matrix for genetic values of **t** traits of an animal; **q** is the number of animals in **A** which is the numerator relationship matrix; $C_{11}, ..., C_{LL}$ are the covariance matrices for the **L** random effects that are correlated across traits but uncorrelated across animals with n_i, the number of sets of each C_{ii}.

The two computing intensive terms are calculated from the Choleski factorization of **C** as $2\,\Sigma\log(\ell_{jj})$ where ℓ_{jj} is the j^{th} diagonal element of the Choleski factor. The Choleski factor can be used to solve for **s** so that $y'Py$ is calculated as $\Sigma y_i'R_i^{-1}y_i - s'r$ where the first term is calculated animal by animal.

The basic steps with sparse matrix techniques are:

1) Symbolically reorder elements of C (once)
2) For each likelihood calculation
   a) update **G** and **R** via simplex and calculate $\log|G|$ and $\log|R|$,
   b) update **C**, **r**, and $\Sigma y_i'R_i^{-1}y_i$ from updated **G**, **R**, and original **y**,
   c) calculate $\log|C|$ and s'r as described above,
   d) check for convergence (based on change in -2Λ).

Times required for these steps were 98 sec to reorder; 44.60 sec to factor, and 1.32 sec to solve (time for a likelihood calculation = 44.60 + 1.32 = 44.92 sec) for a single trait model with direct and maternal genetic effects and maternal permanent effects involving 3,111 animals and 7,303 equations. A traditional derivative method would require inversion of **C** with order 7,303 for each iteration.

A three-trait example (Lucia Albuquerque, personal communication, 1994) introduces problems encountered with multiple trait analyses. The records were milk, fat and protein yields for New York Holsteins with measurements on up to three lactations per cow. The model included animal genetic (9,722) and animal permanent environmental effects (animals with records = 5,706) and management levels (1,509) associated with herd-year-season at initiation of each lactation. The table gives number of equations and computing times for one, two, and three trait analyses:

	Milk	M,F	M,F,P
Equations(no.)	16,937	33,874	50,811
Re-order(sec)	18	61	129
Likelihood(sec)	26	179	594

The advantage of sparsity is illustrated by a similar sample from California including 10,438 animals, 5,877 cows with records and only 225 H-Y-S of freshening. The smaller number of H-Y-S levels resulted in less reorder time and especially less time to calculate -2Λ; 14 sec to reorder for one trait; 103 sec to reorder for three traits and 11 and 143 sec for each likelihood compared to times of 26 and 594 sec for the New York data. The increased time for each calculation of -2Λ combined with many restarts shows that convergence takes a long time with even three traits.

	California		New York	
Number	Milk	M,F,P	Milk	M,F,P
Restarts	1	10	1	10
Λ/Restart	88	400	95	410
Total Λ	88	4000	95	4100
Total time	16.4m	6.6d	41.5m	28.2d

The single trait analyses took a matter of minutes to reach global convergence but the three-trait analyses took about a week for the California sample and about a month for the New York sample due to the time per likelihood calculation and the number of restarts that was needed. The increase in time for calculation of Λ for the New York sample is due to the increase in number of levels of H-Y-S.

Starting values for multiple trait analyses are important with DFREML as illustrated by two analyses with different pairs of traits for the same animals. The first two traits were animal birth weight when born 1) to a young mother and 2) to an older mother. The model included direct and maternal genetic values (with covariance) and maternal permanent environmental effects as some older mothers had more than one calf. Thus, the total number of (co)variances was 15; 3212 animals contributed to relationships, 765 and 1306 calves were born to young and older mothers resulting in 14,676 mixed model equations. Starting values for variances were based on single trait analyses except that a major input error went unnoticed for one maternal genetic variance. A total of 22 restarts (restarts were after 150 simplex rounds or variance of the simplex less than 1.E-6*) was needed before -2Λ changed less than .01 from restart to restart. The pattern of -2Λ after each restart was 11500 plus in turn: 34.23, 29.79, 29.67, 29.28*, 26.82, 25.03, 24.55, 24.43*, 24.15*, 23.87*, 23.56*, 20.99, 18.54, 18.07, 17.97, 17.91, 17.85, 17.72*, 17.42, 17.06, 17.02 and 17.01* when global convergence was assumed. Several times the system seemed on the verge of convergence but would then continue to a better set of estimates.

The similar analyses were with calving ease substituted for birth weight. Calving ease is a trait that is categorically measured which often results in slow convergence. This time the starting values were correctly inputed and only 6 restarts resulted in convergence with consecutive -2Λ of 1007.12, 994.99, 992.04, 987.62, 986.35 and 986.35*. These analyses illustrate some of the frustrations with DFREML for multiple trait analyses and serve to introduce the "art" of DFREML.

The "ART" of DFREML

Convergence

The question of how to proceed most efficiently to find solutions that are globally maximum causes many headaches, results in some degree of doubt about the reliability of DFREML, and is still basically an art form with few established rules. The simplex algorithm is not guaranteed to reach a global minimum (in this case for -2Λ). It may lead to a local minimum. Usually the stopping point after a start is based on the variance of the $n + 1$ log likelihood values retained in the simplex where **n** is the number of parameters. Common stopping points are when $V(-2\Lambda)$ is less than a predetermined value such as 1.E-4, 1.E-6, or 1.E-8. An alternative, based on experience, is to restart after a certain number of simplex rounds or when $V(-2\Lambda)$ is less than the predetermined constant. (Each simplex round requires on average about two likelihood evaluations.) Then -2Λ is examined for improvement from the previous start. If the improvement in -2Λ is less than .01 to .05, then another restart usually results in little additional improvement. Another alternative is based on the previous one but includes an examination of variances as fractions of total variance as well as of correlations. If such proportions do not change in the second decimal, global convergence is likely. Nevertheless, experience as well as such ad hoc guidelines are needed until precise rules are developed. For example, should restarts be limited to a specific number of simplex updates, should restarts be terminated after the variance of simplex has fallen below a pre-determined value, or should some combination be used? What would be the best choices for number of simplex rounds and variances?

The following table shows -2Λ at three convergence levels for 10 samples of milk records with first, second, and third lactations being considered separate traits (Lucia Albuquerque, personal communication, 1994).

-2Λ FOR THREE CONVERGENCE CRITERIA (10 samples)

	Convergence Criterion		
Sample	1.E-4	1.E-6	1.E-9
1	58601.32	58601.20	58601.07
2	55087.71	55087.71	55087.69
3	57122.57	57122.49	57122.49
4	53185.73	53185.71	53185.65
5	52942.48	52942.43	52942.14
6	51778.50	51778.47	51778.46
7	53446.38	53443.84	53443.84
8	50851.60	50851.52	50851.43
9	53778.04	53778.00	53777.97
10	55685.27	55685.24	55685.06

The table illustrates the art of deciding whether global convergence has been reached. For some samples, 1.E-4 and 1.E-6 led to similar -2Λ with 1.E-6 always reaching a smaller (better) value. In other cases, continuing to 1.E-9 resulted in improvement. The importance of differences in -2Λ at the second decimal is difficult to quantify.

Proportional estimates of the variances and correlations for the averages of the same 10 samples at convergence of 1.E-6 and 1.E-9 after many restarts are shown below.

AVERAGES FOR TWO CONVERGENCE CRITERIA

	Convergence Criterion	
Lactations	1.E-6	1.E-9
HERITABILITIES		
01	.35	.35
02	.34	.34
03	.33	.32
GENETIC CORRELATIONS		
(01x02)	.87	.87
(01x03)	.81	.81
(02x03)	.97	.97
ENVIRONMENTAL CORRELATIONS		
(01x02)	.43	.43
(01x03)	.38	.38
(01x03)	.44(.444)	.45(.445)
PHENOTYPIC CORRELATIONS		
(01x02)	.58	.58
(01x03)	.53	.53
(01x03)	.62	.62

To two decimals the averages of proportions were essentially the same. For animal breeding applications even changes in fractional variances from, for example, .30 to .35 are not often important.

Experience has been that 1) for single trait analyses with no imbedded covariances such as the direct-maternal genetic covariance global convergence is usually reached when $V(-2\Lambda)$ is less than 1.E-6, although one restart is a safety measure, 2) for a single trait analysis with a direct-maternal covariance at least one restart is needed and 3) for multiple trait analyses many restarts will be needed with the number dependent on starting values, the complexity of the model, and even the scale of measurements. The multiple trait "rule" is restart, restart, ..., until -2Λ does not change more than about .01.

Boundary Conditions

As with any REML algorithm, solutions outside the parameter space are not estimates. For example, variances must be greater than zero and absolute values of genetic and other correlations must not exceed unity. In addition, eigenvalues of matrices such as R_0 and G_0 which represent environmental and genetic covariance matrices for traits measured on an animal must be positive. As part of the simplex algorithm whenever an update of a solution is not allowed, a large value is assigned to -2Λ which forces a contraction of the simplex update. If necessary, other contractions are forced until the update is allowed. Such contractions are done before the expensive calculation of $\log|C|$ and $y'Py$ so that little time is wasted. Solutions near boundaries, however, often indicate many rounds will be needed as solutions may creep to the boundary of allowed estimates.

Sign of Correlations

The simplex operates by updating current solutions by increasingly smaller fractions of the current solutions. If a starting correlation (covariance) is positive and the optimum solution is negative, the search must pass from positive to negative values of the covariance. Experience indicates that the cross-over requires many rounds of likelihood evaluations.

Rounding in Calculation of -2Λ

A problem that occasionally occurs is that convergence, i.e., variance of -2Λ in the simplex will never be less than 1.E-6. In such cases, that variance typically bounces around at values larger than 1.E-6. Rounding error in calculation of -2Λ from the four components of Λ is likely the reason. The amount of rounding error will be computer and possibly compiler dependent. The potential for rounding error is illustrated by -2Λ values in the range of 190,000 for which V(-2Λ) is to be less than 1.E-6 or 1.E-8 at convergence.

Another experience also may be due to rounding error. At least one analysis has shown a cyclic fluctuation in -2Λ such as 24470, 24490, 24470, ... which insures that V(-2Λ) is large and convergence based on V(-2Λ) will not be attained. Examination of the solutions showed only slight differences even though the -2Λ for each set of solutions were quite different. A possible explanation is that parts of the likelihood involve logs of small eigenvalues which are then multiplied by a number such as the number of animals.

Binomial data with values of 0 and 1 or 1 and 2 have led to the problem described in the previous paragraph. When convergence has not been attained, two approaches have been followed. Multiplying the binomial values by 100 sometimes seems to lead to better numerical properties. Another alternative has been to change to a sire model rather than to continue with an animal model.

Starting Values

The importance of starting values depends primarily on whether the analysis contains covariance terms. For a single trait analysis, the sum of components at the start should be reasonable, i.e., less than the raw variance. At least one analysis failed to reach convergence when, by oversight, the starting variances were all several times the true variances. With direct-maternal genetic covariance included for a single trait, choice of correct sign of the covariance is important as discussed earlier. The covariance should not be started as zero because the steps of the simplex algorithm are proportions of the previous solutions. A starting zero will remain zero.

Multiple trait analyses take more time per round, more rounds to simplex convergence and, usually, many restarts to attain global convergence; thus good starting values are important. One suggestion is: 1) do single trait analyses to determine variances and within-trait direct maternal covariances, 2) start with across-trait covariances corresponding to moderate correlations and the better guess of positive or negative sign while holding variances from 1) constant (an option in the MTDFREML program); and then 3) let all (co)variance elements vary in the simplex with the prospect of several restarts.

Conclusions

Derivative-free REML with sparse matrix methods based on Henderson's mixed model equations has expanded the magnitude of single and multiple trait analyses to obtain REML estimates of variances and covariances. Single trait analyses converge quickly. The "art" of DFREML mainly involves rules for reducing time to global convergence for multiple trait analyses. Optimum starting values and restart strategies have not been determined, although obvious ad hoc rules have been evolving. Restarts to insure convergence to a global maximum for Λ (or minimum for -2Λ) are mandatory for multiple trait analyses. Help is needed 1) to develop an improved updating algorithm, 2) to determine starting strategies for multiple trait analyses, and 3) to design a general method for restarting to obtain most efficiently solutions that have converged to the global maximum of the likelihood given the data.

References

Boldman, K. G. and L. D. Van Vleck. 1990. Effect of different starting values on parameter estimates by DF-REML and EM-REML in an animal model with maternal effects. J. Anim. Sci. 68(suppl. 2):71 (abstr.).

Boldman, K. G. and L. D. Van Vleck. 1991. Derivative-free restricted maximum likelihood estimation in animal models with a sparse matrix solver. J. Dairy Sci. 74:4337.

Boldman, K. G., L. A. Kriese, L. D. Van Vleck, and S. D. Kachman. 1993. *A Manual for Use of MTDFREML.* A set of programs to obtain estimates of variances and covariances. USDA-ARS, Roman L. Hruska U.S. Meat Animal Research Center, Clay Center, NE.

Chu, E., A. George, J. Liu, and E. Ng. 1984. SPARSPAK: Waterloo sparse matrix package user's guide for SPARSPAK-A. CS-84-36, Dept. Computer Sci., Univ. Waterloo, Waterloo, ON, Canada.

George, A., J. Liu, and E. Ng. 1980. User guide for SPARSPAK: Waterloo sparse linear equations package. CS-78-30, Dept. Computer Sci., Univ. Waterloo, ON, Canada.

Graser, H. -U., S. P. Smith, and B. Tier. 1987. A derivative-free approach for estimating variance components in animal models by restricted maximum likelihood. J. Anim. Sci. 64:1362.

Groeneveld, E. and M. Kovac. 1990. A note on multiple solutions in multivariate restricted maximum likelihood covariance component estimation. J. Dairy Sci. 73:2221.

Harville, D. A. 1977. Maximum likelihood approaches to variance component estimation and to related problems. J. Amer. Stat. Assoc. 72:320.

Henderson, C. R. 1963. Selection index and expected genetic advance. In: Statistical Genetics in Plant Breeding. NAS-NRC publication 982.

Henderson, C. R. 1984. *Application of linear models in animal breeding*. U. Guelph, Guelph, ON, Canada.

Meyer, K. 1988. DFREML. A set of programs to estimate variance components under individual animal model. J. Dairy Sci. 71(suppl. 2):33.

Meyer, K. 1989. Restricted maximum likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. Genet. Sel. Evol. 21:317.

Meyer, K. 1991. Estimating variances and covariances for multivariate animal models by restricted maximum likelihood. Genet. Sel. Evol. 23:67.

Nelder, J. A. and R. Mead. 1965. A simplex method for function minimization. Computer J. 7:308.

Patterson, H. D. and R. Thompson. 1971. Recovery of inter-block information when block sizes are unequal. Biometrika 58:545.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. 1989. *Numerical recipes*. Cambridge University Press, New York.

Quaas, R. L. 1976. Computing the diagonal elements and inverse of a large numerator relationship matrix. Biometrics 32:949.

Searle, S. R. 1979. Notes on variance component estimation: A detailed account of maximum likelihood and kindred methodology. Paper BU-673-M, Biometrics Unit, Cornell Univ.

Smith, S. P. and H. -U. Graser. 1986. Estimating variance components in a class of mixed models by restricted maximum likelihood. J. Dairy Sci. 69:1156.

Stewart, G. W. 1994. Gauss, statistics, and Gaussian elimination. Keynote address, these proceedings.

Dual Space Algorithms for Designing Space-Filling Experiments

William D Heavlin, G Paul Finnegan
Advanced Micro Devices, MS 152
POBox 3453, Sunnyvale, California 94088-3453
Internet: bill.heavlin@amd.com

Key words: computer experiments, D-optimality, goodness of fit, Latin hypercubes

Abstract: For generating response surface designs, most general purpose ("D-optimal") algorithms work point by point in the design domain. We introduce a class of algorithms operating in the dual, factor/column space. Their basic operations exchange, randomly and systematically, the rows of certain columns (factors) with respect to the rows of other columns.

This dual space approach is especially suitable for designing computer experiments of Latin hypercube type. The experimenter can embed two- and three-level response surface designs, both to match a calibration subset and to achieve high efficiency. More centrally, the experimenter explicitly chooses the number of factor levels and their frequencies, ideal both for considering model-free goodness-of-fit and for establishing interpolation grids.

1. Computer experiments

Complicated physical phenomena are increasingly well modeled by computer simulators. The underlying physical theory usually involves two-to-four dimensional differential equations with boundary conditions, key complications consist of multiple materials, their interfaces and geometrical structures. Important methods encompass automatic grid generators, parallel computing algorithms, finite element analysis, and empirical metrology and calibration procedures. Typical simulator applications include verification and optimization of product designs and policies, diagnosis of problems and opportunities, evaluation of difficult-to-measure constructs, development of predictive models, and estimates of distributions.

Experiments using computer simulators, so-called computer experiments, are a focus of statistical methods research. Among their special considerations are their ability to repeat perfectly, the increased feasibility to run larger experiments, and the opportunity to fit richer, more nonparametric models. Modeling approaches include kriging (Matheron [1971], Sacks et al [1989]), nonparametric regression (Friedman [1991]), and neural networks (Cheng and Titterington [1994]). This work is shaped by target applications: Sacks et al (1989) predict, then optimize analog integrated circuit performance. Friedman (1991) emphasizes graphical visualization and decomposition. Several authors perturb simulator inputs to project output distributions. Their methods range from Monte Carlo sampling (Kibarian and Strojwas [1991]), low-order moment estimation (Zaino and D'Errico [1988]), and Latin hypercube sampling (McKay et al. [1979]).

This paper is on designing computer experiments, in particular using Latin hypercubes (LHCs). When introduced by McKay et al (1979), LHCs were constructed by random mechanisms, and have since been shown to be more efficient for distribution estimation than Monte Carlo sampling (Stein [1987], Owen [1992a]). LHCs' advantage is further increased by constructing them using orthogonal arrays (Owen [1992b], Tang [1993]).

An alternative computer experiment design approach is that of Sacks et al (1989), who introduce a class of optimal experiments based on a kriging model, in the sense of minimizing integrated mean square error. Figure 1 shows the scatterplot matrix of their 32-run 6-factor design. Note that each projection into two dimensions shows a characteristic five-spot X-pattern. Many researchers have found this pattern objectionable, preferring the symmetry of Latin hypercubes.

2. Problem Statement

Like many others, our ultimate application is distribution estimation. The economics of simulation motivate our approach. Simulations are relatively slow, on the order of 12-24 hours each. This encourages us to build an intermediate model, one from which we can interpolate other values. Also, at certain points in the design domain we have empirical measurements, whose configuration forms a conventional $2^{F-p}3^p$ response surface design. To these we need to match their corresponding simulations, in order to calibrate the model correctly. The empirical measurements are also precious, and the time it takes to develop them ultimately bounds the number of simulations we can perform.

To summarize, the particular characteristics of our computer experiment are the following: (1) Computer simulations are time-consuming, hence precious, and efficient designs are therefore desirable. (2) The computer experiment is used to establish an interpolation grid, from which an easy-to-evaluate model can be developed. (Observe that the application of Latin hypercube sampling, by which a simulator is evaluated in order to estimate the distribution of an output parameter, is moved outside our scope. We can, of course, apply Latin hypercube sampling to our interpolated model.) (3) Some of the computer simulations are fixed a priori (to match empirical measurements for calibration, perhaps to improve the interpolation grid), and we would like take advantage of these runs. (4) The size of the experiment is small to moderate — for definiteness, say 50-100 runs of about 8-10 factors. (5) Beyond design optimality, we would like to preserve some sensitivity to detecting model lack of fit.

3. Optimal Experiments and Design Repair

Much of the theory of optimal designs is based on conventional linear models, with homogeneous, independent errors. This literature has two themes. By one theme, with respect to a particular model, one defines criteria by which one can compare designs. These are usually functions of the coefficients' variance-covariance matrix; the most common is the determinant, the so-called D-optimality criterion. By theme two, the design domain, in principle continuous, is reduced to a finite set. For example, with one factor, the optimal design of a linear model is well known to concentrate all points at the extremes of the feasible range. Similarly, optimal designs for quadratic models concentrate all design points at three levels. Atkinson and Donev (1993) give a contemporary account of optimal design literature. In practice, for computer experiments, the limited variety of points in the design domain has made conventional optimal designs unattractive.

Our basic approach adapts the columnwise D-optimal algorithm of Heavlin and Finnegan (1993). The "design repair" algorithm presented therein uses the D-optimal criterion (theme one, above), but not the restricted design domain (theme two). Instead, the experimenter chooses each factor's levels, and the frequency with which they are used. For computer experiments of the Latin hypercube type, with n runs, this means the levels of each factor are the values $1,2,\ldots,n$; equal spacing is used to improve the interpolation grid.

The design repair approach also uses conventional linear models, and homogeneous, independent errors. Its natural domain of applicability is sequential batch processes, e.g. semiconductor manufacturing. Applications include assigning interacting covariates, adapting experiments to lost experimental units (e.g. broken silicon wafers), designing responses with partially overlapping factor sets, and allocating noise-factor batch positions. For conventional response surface designs, design repair has proven useful for finding partially balanced incomplete block designs, combining mixture and nonmixture factors, creating level-balanced response surface designs, and constructing loss resistant experimental designs.

Design repair's primary data structures are two partial design matrices, W and X, and one model. Both W and X have n rows, and F_W and F_X columns respectively. In addition, for certain problems we wish to include certain experiments, certain complete rows. We denote these rows by WX_0. Let W_i (X_i) denote the ith row of W (X). Let π denote a permutation of the row indices $1,2,\ldots,n$. We would like to form the design matrix WX^{π}, whose ith row is W_i and $X_{\pi(i)}$, and which includes the a priori rows WX_0, that is,

$$WX^{\pi} = \begin{bmatrix} W \quad X_{\pi} \\ \hline WX_0 \end{bmatrix},$$

where X_{π} denotes the $n \times F_X$ matrix whose ith row is $X_{\pi(i)}$. From WX^{π} we can develop a model matrix M^{π}, whose ith row is $M_i^{\pi} = m(WX_i^{\pi})$. For example, were W and X both one column matrices, and our desired model a full quadratic, the $m(u_1,u_2)$ returns the row vector $(1,\ u_1,\ u_2,\ u_1u_2,\ u_1^2,u_2^2)$, corresponding to the constant, two linear, one interaction, and two quadratic terms. Hence, M^{π} has the form

$$M^{\pi} = \left[\begin{array}{c|c} W \qquad X_{\pi} & \text{higher} \\ \text{-- -- -- -- -- --} & \text{order} \\ WX_0 & \text{terms} \end{array}\right].$$

The design repair algorithm works to find the best π, or at least a good one, so that we can estimate by least squares the linear coefficients of M^{π}. We choose the D-optimality criterion, for which larger values are better:

$$\begin{aligned} D(M) &= \ln(\det(M^{\mathrm{T}}M)), \text{ for } M \text{ non-singular}, \\ &= -\infty, \qquad\qquad\qquad\qquad \text{otherwise}. \end{aligned}$$

With a quantitative measure (D) of a good design specified, the design repair algorithm is easily imagined. It comes in two parts, a *random* starting point, called R-step, and a deterministic search over *exchanges* or transpositions of pairs of elements in π, called E-step.

R-step: π is selected at random from all possible permutations, WX^{π} and M^{π} constructed, and $D(M^{\pi})$ evaluated. This is repeated for n_R iterations, keeping track of the best π. As the number of iterations increases, discovery of better permutations becomes less likely and R-step becomes inefficient. This motivates E-step.

E-step: As a starting point, E-step uses the best permutation found from R-step, say π_R. All $\binom{n}{2}$ combinations formed by exchanging a pair of indicies are then considered, that with the largest D-value selected. In this way, E-step is repeated until no further improvement from pairwise exchanges of indices (rows of X^{π}) is found.

Specification of n_R: For the optimum number of R-steps, n_R^*, Heavlin and Finnegan (1993) develop an heuristic and approximate relationship: $log_{10}(n_R^*) \doteq -0.71 + 2.12\, log_{10}(n)$. In one region of interest, n about 50, this implies $n_R^* \doteq 800$.

4. Computer Experiment Test Case

To use the design algorithm, one must specify the matrices W and X, and an appropriate model. For computer experiments, one usually needs to apply the design repair algorithm several times in series, building up the columns of the design in stages. Let W^j, X^j, and m^j denote these objects for the jth application of design repair. Denote by $DR(W^j,X^j,m^j)$ the solution from the jth step. Three issues need addressing:

1. Path: $W^1 = X^1 = (1,2,\ldots,n)^{\mathrm{T}}$ seems natural, as does $W^{j+1} = DR(W^j, X^j, m^j)$. How should X^j be selected for $j \geq 2$? The fastest route is $X^j = W^j$, which we call "doubling." This allows us to start initially with a column matrix, then obtain a two-column design, then four, then 8, and so on. The alternative is to choose $X^j = X^1$ for all j. This builds up the design slowly, one column at a time. This path we call "add one."

2. Bases: Should the model be described as a polynomial, or are there useful alternatives, such as using terms of a Fourier series?

3. Models: What model, in particular which interactions, should be specified? At one extreme, one can specify a purely additive model, with no interactions among the factors; at the other extreme, one might pose as large a set of interactions as feasible.

As a test case, we develop an 8-factor, 51-run Latin hypercube. There are several reasons for this choice. The size of this experiment is large enough to be practical, yet small enough for design repair to handle reasonably. 51 runs allow us to specify $W^1 = (-1, -0.96, -0.92, \ldots, 0, \quad 0.04, \quad \ldots, +1)^{\mathrm{T}}$. Finally, Tang (1993) has published scatterplot matrices for a 49-run 8-factor LHC constructed using orthogonal arrays, giving us a good standard for comparison. To facilitate comparisons, we use no WX_0 matrix.

For this exercise, we follow both the doubling path, and the add-one path. For bases, we use a 7-degree (orthogonalized) polynomial, whose terms before orthogonalization correspond to w, w^2, w^3, w^4, w^5, w^6, and w^7; these seven columns comprise W^1. As an alternative basis, we also consider seven terms of a Fourier series, corresponding to w, $sin(2\pi w)$, $cos(2\pi w)$, $sin(4\pi w)$, $cos(4\pi w)$, $sin(8\pi w)$, $cos(8\pi w)$, also orthogonalized. To enhance comparability, for these four designs, we choose additive models, with no interactions; W^2 is the same in all constructions, the attractive result of a design repair construction using the seven-term Fourier basis and a high-order interaction model.

Judging from scatterplot matrices, the most satisfying design is that using the polynomial basis and add-one path (figure 2), comparable to Tang's figure 3 of an orthogonal array-based LHC. Space limitations prohibit showing scatterplot matrices of the other bases and paths, but the scatterplot matrices of both add-one constructions are more satisfying, with points well spread out and no large area unoccupied, than those from the doubling path. (This agrees with the authors' experiences in other computer experiment applications.) In both cases, the polynomial constructions are somewhat more pleasing than those using the Fourier series.

Figure 3 is the scatterplot matrix of the 51-run 7-factor design repair construction. Like that in figure 2, it is the result of the add-one path and the 7-term polynomial basis. Unlike figure 2, it uses a series of rich models: W^3, a full six-order model (81 terms); W^4, a full fifth-order model (124 terms); W^5, a full third-order model, plus all fourth-order terms involving the fifth factor, plus pure quartic terms for all five factors (127 terms); W^6, a full cubic model, plus all pure fourth-order terms, plus all mixed interactions involving the sixth factor (99 terms); and W^7, a full cubic model, plus pure quartic terms (77 terms). These models have more terms than runs; the D-criterion is modified to $ln(det(M^{\mathrm{T}}M + \lambda I))$, with $\lambda = 0.1$.

Under the constraints of the LHC margins, figure 3 shows an X-pattern similar to that of Sacks et al (1989). One might speculate that the kriging model is related to interaction-rich models. An alternate interpretation is that the design repair approach to LHC construction should be applied only for additive models.

5. Conclusions

The design repair algorithm can construct Latin hypercube designs successfully. Conditions where this is appropriate are listed in section 2; the key ingredients are a design of moderate scope with some particular requirements. Based on reviewing scatterplot matrices of the resulting designs, polynomial models work at least as well as the alternatives. The add-one path allows the models to be specialized to each step of construction; for this reason, it is not unexpected that add-one designs have better esthetic properties than designs based on the doubling path.

A well recognized analogy is on one hand with two-level factors, linear models, and resolution III projection properties and, on the other hand, with multilevel factors, additive models, and two-dimensional projections (called strength 2). For this reason, one might anticipate that additive models would give appealing scatterplots matrices, which are merely graphical strength 2 assessments. The similarity of X-patterns both in Sacks et al (1989) and figure 3's interaction-rich Latin hypercube construction is more tantalizing, perhaps pointing to some connection between the two approaches for high-dimensional designs.

6. References

Atkinson, AC, and Donev, AN (1992). *Optimum Experimental Designs.* Clarendon Press, Oxford.

Cheng, B, and Titterington, DM, (1994), "Neural networks: A review from a statistical perspective," with discussion, *Statistical Science, 9,* 1, pp2-54.

D'Errico, JR, and Zaino, NA Jr, (1988), "Statistical tolerancing using a modification of Taguchi's method," *Technometrics, 30,* pp397-405.

Friedman, JH, (1991), "Multivariate adaptive regression splines," with discussion, *Annals of Statistics, 19:1,* pp1-141.

Heavlin, WD, and Finnegan, GP, (1993), "Adapting experiments with sequentially processed factors," *ASA Proceedings, Section on Physical and Engineering Science,* August, San Francisco.

Kibarian, JH, and Strojwas, AJ, "Using spatial information to analyze correlations between test structure data," *IEEE Trans. on Semiconductor Manufacturing, 4,* August, pp 219-225.

Matheron, G (1971), *The Theory of Regionalized Variables,* Centre of Morphologie Mathématique de Fontainbleau.

McKay, MD, Conover, WJ, and Beckman, RJ, (1979), "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics, 21:* pp239-245.

Owen, AB, (1992a), "A central limit theorem for Latin hypercube sampling," *Journal of the Royal Statistical Society, Series B, 54,* 2, pp541-551.

Owen, AB, (1992b), "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica, 2:2,* pp439-452.

Sacks, J, Welch, WJ, Mitchell, TJ, and Wynn, HP (1989), "Design and analysis of computer experiments," with discussion, *Statistical Science, 4:*4, pp409-435.

Stein, M (1987), "Large-sample properties of simulations using Latin hypercube sampling," *Technometrics, 29,* pp143-151.

Tang, B, (1993), "Orthogonal array-based Latin hypercubes," *Journal of the American Statistical Association, 88:*424, pp1392-1397.

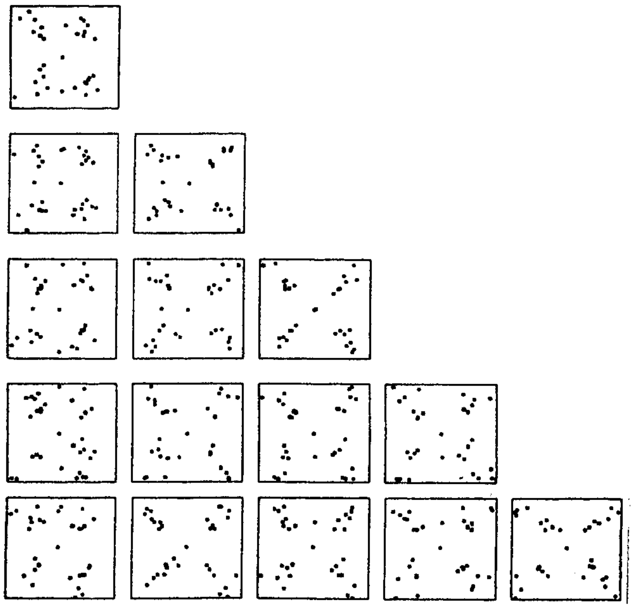


Figure 1. Scatterplot matrix of the 6-factor computer experiment of Sacks et al (1989). The optimality criterion is minimum integrated mean square error; the model a kriging one.

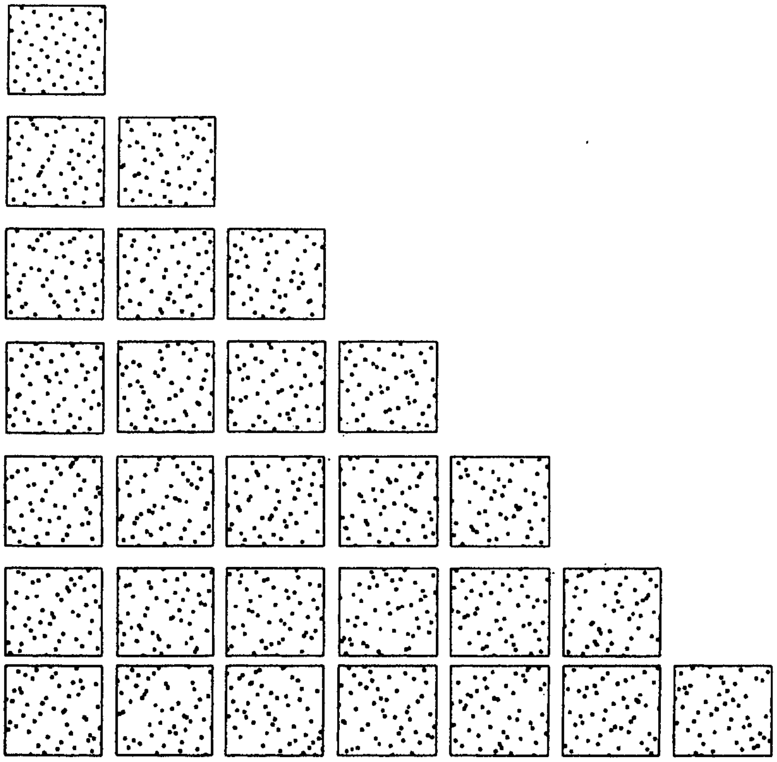


Figure 2. Scatterplot matrix of a 51-run 8-factor Latin hypercube using the design repair algorithm. The model is an additive 7-degree polynomial (no interactions); the factors are added one-by-one.

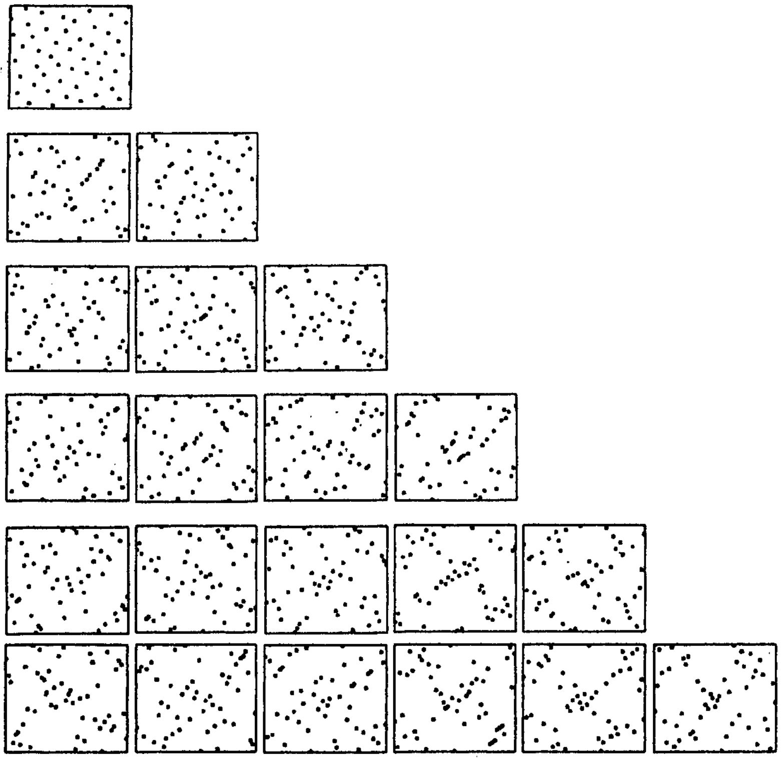


Figure 3. Scatterplot matrix of a 51-run 7-factor Latin hypercube using the design repair algorithm. The model specifies high order interactions terms; the factors are added one-by-one.

figure 3 shows an X-pattern similar to that of Sacks et al (1989). One might speculate that the kriging model is related to interaction-rich models. An alternate interpretation is that the design repair approach to LHC construction should be applied only for additive models.

5. Conclusions

The design repair algorithm can construct Latin hypercube designs successfully. Conditions where this is appropriate are listed in section 2; the key ingredients are a design of moderate scope with some particular requirements. Based on reviewing scatterplot matrices of the resulting designs, polynomial models work at least as well as the alternatives. The add-one path allows the models to be specialized to each step of construction; for this reason, it is not unexpected that add-one designs have better esthetic properties than designs based on the doubling path.

A well recognized analogy is on one hand with two-level factors, linear models, and resolution III projection properties and, on the other hand, with multilevel factors, additive models, and two-dimensional projections (called strength 2). For this reason, one might anticipate that additive models would give appealing scatterplots matrices, which are merely graphical strength 2 assessments. The similarity of X-patterns both in Sacks et al (1989) and figure 3's interaction-rich Latin hypercube construction is more tantalizing, perhaps pointing to some connection between the two approaches for high-dimensional designs.

6. References

Atkinson, AC, and Donev, AN (1992). *Optimum Experimental Designs.* Clarendon Press, Oxford.

Cheng, B, and Titterington, DM, (1994), "Neural networks: A review from a statistical perspective," with discussion, *Statistical Science, 9,* 1, pp2-54.

D'Errico, JR, and Zaino, NA Jr, (1988), "Statistical tolerancing using a modification of Taguchi's method," *Technometrics, 30,* pp397-405.

Friedman, JH, (1991), "Multivariate adaptive regression splines," with discussion, *Annals of Statistics, 19:1,* pp1-141.

Heavlin, WD, and Finnegan, GP, (1993), "Adapting experiments with sequentially processed factors," *ASA Proceedings, Section on Physical and Engineering Science,* August, San Francisco.

Kibarian, JH, and Strojwas, AJ, "Using spatial information to analyze correlations between test structure data," *IEEE Trans. on Semiconductor Manufacturing, 4,* August, pp 219-225.

Matheron, G (1971), *The Theory of Regionalized Variables,* Centre of Morphologie Mathématique de Fontainbleau.

McKay, MD, Conover, WJ, and Beckman, RJ, (1979), "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics, 21:* pp239-245.

Owen, AB, (1992a), "A central limit theorem for Latin hypercube sampling," *Journal of the Royal Statistical Society, Series B, 54,* 2, pp541-551.

Owen, AB, (1992b), "Orthogonal arrays for computer experiments, integration and visualization," *Statistica Sinica, 2:2,* pp439-452.

Sacks, J, Welch, WJ, Mitchell, TJ, and Wynn, HP (1989), "Design and analysis of computer experiments," with discussion, *Statistical Science, 4:*4, pp409-435.

Stein, M (1987), "Large-sample properties of simulations using Latin hypercube sampling," *Technometrics, 29,* pp143-151.

Tang, B, (1993), "Orthogonal array-based Latin hypercubes," *Journal of the American Statistical Association, 88:*424, pp1392-1397.

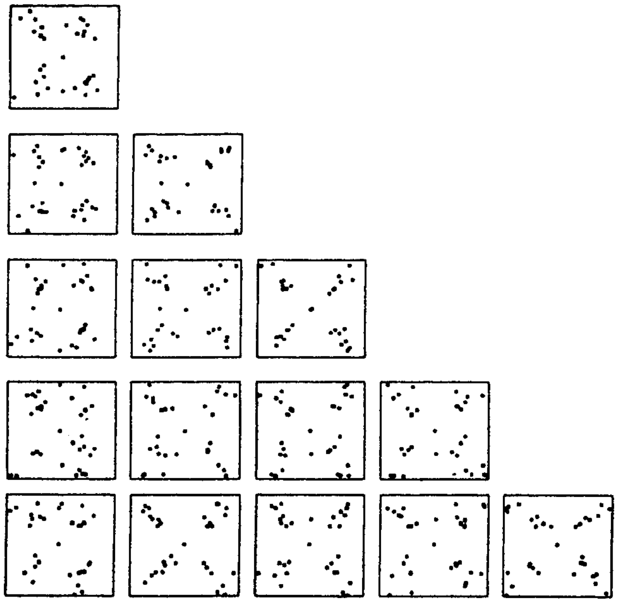


Figure 1. Scatterplot matrix of the 6-factor computer experiment of Sacks et al (1989). The optimality criterion is minimum integrated mean square error; the model a kriging one.

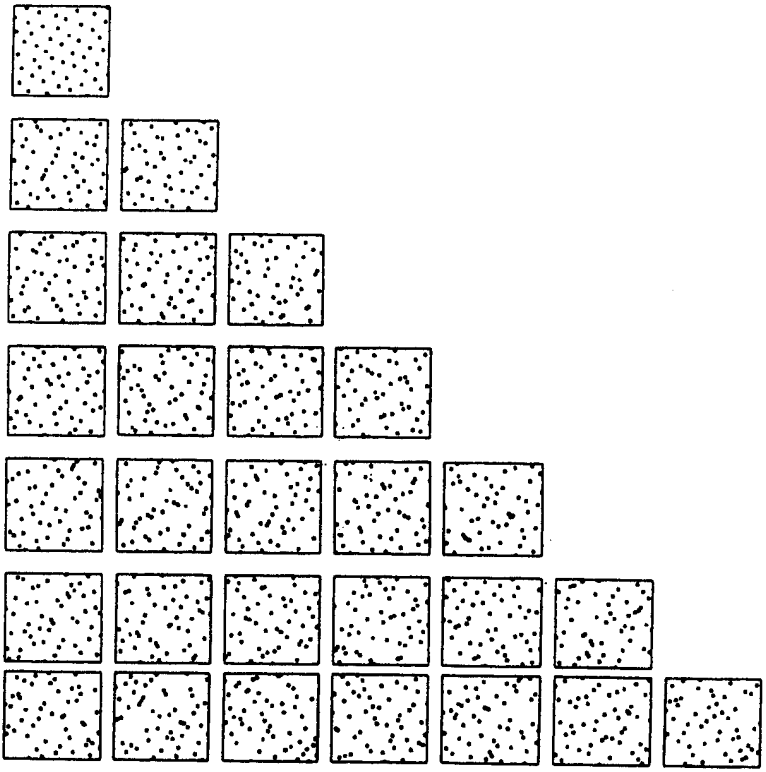


Figure 2. Scatterplot matrix of a 51-run 8-factor Latin hypercube using the design repair algorithm. The model is an additive 7-degree polynomial (no interactions); the factors are added one-by-one.

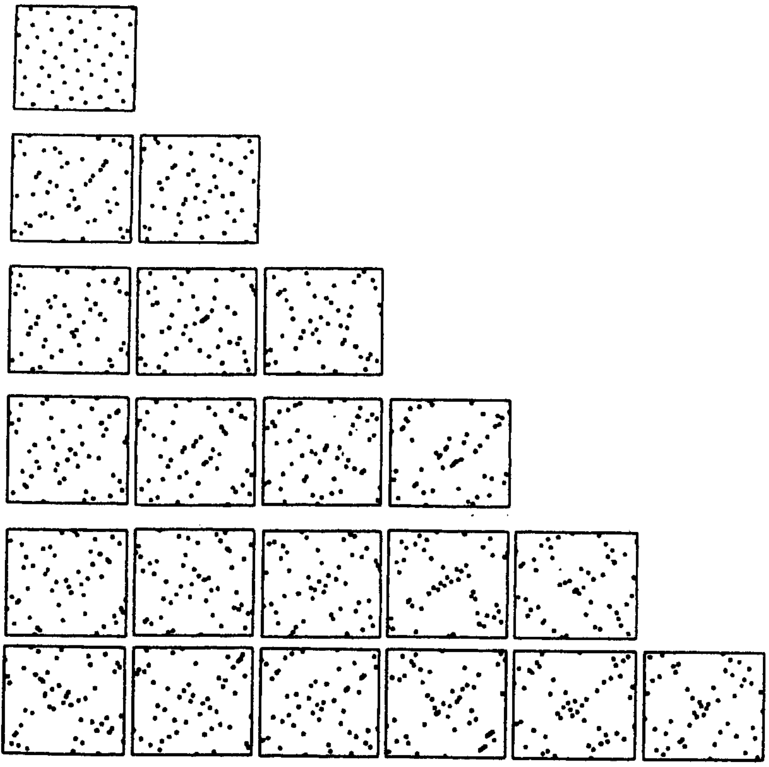


Figure 3. Scatterplot matrix of a 51-run 7-factor Latin hypercube using the design repair algorithm. The model specifies high order interactions terms; the factors are added one-by-one.

Large Sampling Plans On The Sphere

Jason. J. Brown
Department of Statistics
222 Math Sciences Columbia, MO, 65211

Abstract

In recent years, modeling spatial processes on the sphere (*e.g.,* mining, oil exploration, forestry, pollution, ozone levels, etc.) has become more abundant. But through it all, there has been no generally accepted global sampling plan and none for which a central limit theorem (CLT) nor resampling algorithm has been formulated. Some of the global sampling plans that have been used are either derived from experimental design methods or geographical methods. In this paper, we outline each of the above types of sampling plans, describing their strengths and weaknesses, and then describe a global sampling plan called a stratified spherical sampling plan (Brown [1993a]) for which a CLT has been proved (Brown [1993b]) and bootstrap algorithm has been developed and strong uniform consistency of the sample mean has been proved (Brown [1993c]).

Background

Spherical data arises in many disciplines: astrophysics (star clusters), health sciences (MRI, contaminants), geology (oil, earthquakes), meteorology (ozone, pollution), and geography (water levels, coast line) just to name a few. Sampling plans play a major role in characterizing a random field and the dependence structure of statistics defined on the random field. In particular, creating confidence intervals and conducting hypothesis tests on statistics are directly related to the sampling plan.

Unfortunately there is no generally accepted way to gather spherical data. In particular, we would like a global sampling plan upon which we can prove a CLT and/or create a resampling algorithm. Up until 1993, the only sampling plan for which a CLT has been proved is for the continually indexed sphere (Leonenko and Yadrenko [1979]).

The most common way to prove a CLT for dependent data is to use a characteristic of the random field known as stationarity (translation invariance) and the big-block, little-block methodology and α-mixing to reduce the problem to the iid setting. Using these ideas when the sample size n is very large, if the small blocks are small in size compared to the big blocks, but still large enough to separate the big blocks by a substantial amount, then the big blocks act almost independently (α-mixing) while the small blocks are negligible compared with the big blocks. The stationary insures that the statistic defined on the big blocks are iid. Note that when working with spherical data, we assume that the random field is isotropic or rotation and translation invariant.

Resampling algorithms are usually employed when interest is in a parameter θ of some distribution F and the estimate of θ is cumbersome and the calculations of the distribution of the estimate are intractable. Usually one wishes to create confidence intervals for θ and/or do hypothesis testing on θ; a resampling algorithm estimates the true distribution of the statistic and this estimated distribution is used in the inference.

In this setting, we collect data $(X_1, X_2, \ldots, X_n) = \vec{X}_n$ from F, use a statistic $t_n = t_n(\vec{X}_n)$ that estimates θ, and determine the distribution of t_n. The field of resampling tries to estimate the distribution of t_n by reusing the data at hand to create more samples and hence more statistics. We investigate the bootstrap here, but there are many other resampling methods.

In 1979, Efron described an resampling method called the bootstrap for iid data. This method is paraphrased as follows: from data X_1, X_2, $\ldots$, X_n, calculate the empircal distribution function of the data $\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{I}\{X_i \leq x\}$. Resample n observations iid from $\hat{F}_n(x)$ to create a bootstrap sample $\vec{X}_n^{b*} = (X_1^*, X_2^*, \ldots, X_n^*)$. Calculate $t_n^{b*} = t_n(\vec{X}_n^{b*})$ a bootstrap statistic of t_n. Repeat this procedure B times and use the distribution of the t_n^{b*}s as an estimate of the distribution of t_n.

In fact, the true bootstrap estimate of the mean of t_n is $\mu_{Boot} = E_{\hat{F}_n}\{t_m^* | \vec{X}_n\}$ which is estimated by

$$\hat{\mu}_{Boot} = \frac{1}{B}\sum_{b=1}^{B} t_m^{b*} = \bar{t}_n^*,$$

and the true bootstrap estimate of the variance of t_n is

$\sigma^2_{Boot} = V_{\hat{F}_n}\{t^*_m|\vec{X}_n\}$ which is estimated by

$$\hat{\sigma}^2_{Boot} = \frac{1}{B-1}\sum_{b=1}^{B}(t_n^{b*} - \bar{t}_n^*)^2.$$

Carlstein [1986] extended Efron's bootstrap to time-series data by creating the non-overlapping blockwise bootstrap. His method creates blocks with identical joint distributions (due to the stationarity of the time-series), the blocks are then treated as the X_i in the iid setting. In particular, from the n observations of the time-series, let $k = n/l$ and $B_i = (X_{il+1}, X_{il+2}, \ldots, X_{(i+1)l})$ be the ith block of l observations. The stationarity insures that the statistics defined on the k blocks B_i have the same joint distribution. We resample k blocks from $\hat{F}_l$, the empirical disrtibution function of the length l blocks, and join them together to form a bootstrap time-series. Calculate the statistic on the bootstrap time-series and repeat B times.

Künsch [1989] extended this method to the overlapping blocks case. Here there are $n-l+1$ blocks of length l, $B_i = (X_{i+1}, X_{i+2}, \ldots, X_{i+l})$, but the blocks now overlap, whereas before they did not. We again resample k blocks from this collection and repeat the above process. In comparison to the nonoverlapping case, this method reduces the variance of the estimate of variance of the sample mean by 1/3.

Therefore, in order to prove a CLT and create a blockwise resampling algorithm it is necessary for a global sampling plan to have separating blocks for the big-block, little-block theory and repeating patterns for the isotropy of the random field.

Sampling Plans

There are two basic approaches for creating global sampling plans: experimental design considerations and geographical considerations. The experimental design approach does not necessarily generate designs which have repeating patterns that are necessary in a blockwise resampling algorithm, but they have design optimality properties for certain models. On the other hand, geographical sampling methods are used to create designs with repeating patterns, but do not have the design optimality property. Geographical sampling plans fall into one of two types: polyhedral tessellations and map-projections.

Experimental Designs

The experimental design approach begins with the following setup: Consider the specific model with k variables $x_1, \ldots, x_k$, $p = \frac{1}{2}(k+1)(k+2)$ unknown parameters β, and error term ϵ with mean 0 and variance σ^2,

$$y = \beta_0 + \sum_{i=1}^{k}\beta_i x_i + \sum_{i=1}^{k}\beta_{ii}x_i^2 + \sum_{i=1}^{k-1}\sum_{j=i+1}^{k} x_i x_j + \epsilon.$$

Let $(x_{j1}, \ldots, x_{jk})$ be a design point in the region of operability O and X be the design matrix containing rows $f(x) = (1, x_1, \ldots, x_k, x_1^2, \ldots, x_k^2, x_1x_2, \ldots, x_{k-1}x_k)$. The moment matrix is then $M_X = X'X/n$ and the prediction variance is $V\{\hat{y}(x)\} = f(x)M_X^{-1}f'(x)\sigma^2/n$. If we let R be the modeling region and $\mu(\cdot)$ be a uniform measure over R with total measure 1, then we can then choose design points so as to minimize any one of the criterions in Table 1.

I-optimal	:	$\min \operatorname{trace}\{MM_X^{-1}\}$
		where $M = \int_R f'(x)f(x)d\mu(x)$
A-optimal	:	$\min \operatorname{trace}\{M_X^{-1}\}$
D-optimal	:	$\min \det\{M_X\}^{-1/p}$
E-optimal	:	$\min \max_i e_i(M_X^{-1})$
		where $e_i(\cdot)$ are the eigenvalues
G-optimal	:	$\min \max_R V\{\hat{y}(x)\}$

Table 1: Optimality Criterion

In 1993, Hardin and Sloane introduced a computer algorithm called GOSSET that used a modification of the pattern search method of Hooke and Jeeves [1961]. The algorithm uses the gradient of a differential function to find the minimum and hence it is able to find I-, A-, or D- optimal designs, not E- and G-optimal. It can be used with very complicated O and R (balls, cube, hyperplanes, and intersections and unions).

We are interested in balls. Unfortunately, when the sample size gets large, the sampling plans created by GOSSET do not have repeating patterns.

Polyhedral Tessellations

The polyhedral tessellation sampling plans usually start from one of the 5 platonic solids: tetrahedron, hexahedron (cube), octahedron, dodecahedron, and icosohedron. The solid is then inscribed in a sphere and its edges are projected onto the sphere as great arcs. Most of the tessellations then apply the alternate method of Gasson [1983]. His method states that each spherical triangle can be recursively subdivided into four subtriangles by placing vertices at the midpoint of an edge and joining the new vertices. This method has relation to geodesic domes (Popko [1968]).

Dutton [1989] uses the octahedron as a basis for a quaternary triangular mesh (QTM). Here each face is a spherical triangle and is recursively subdivided using the alternate method. Dutton points out that one can get 1 meter resolution in 21 recursions. Goodchild and Shiren [1989] provided a conversion to the latitude-longitude scale since in this setting the "base" edges for the octahedron and its subdivisions are parallel to the equator. Unfortunately this method does not subdivide into equal area cells nor equal shapes.

Wickman, Elvers, and Edvarson [1974] use the dodecahedron as basis for their method. Here, each face is a pentagon and is first subdivided into 5 isosceles triangles. One then recursively subdivides the isosceles triangles by the alternate method. The sphere can be subdivided into equal area pieces, but they have different shapes.

Fekete [1990] uses the icosohedron as basis for a sphere quadtree (Samet [1984]). In his approach, each face is a triangle and the alternate method is applied to each. This quadtree does not subdivide into equal area pieces, but there is less distortion of size and shape than the QTM method.

White, Kimerling, and Overton [1992] use the truncated icosohedron as a basis for their method. The truncated icosohedron has faces that are both pentagons and hexagons and is the common design for soccer balls. They begin by decomposing the pentagons into 5 triangles and the hexagons into 6 triangles. They then apply the alternate method subdivision on each triangular face. Their method also does not subdivide into equal area pieces, but there is less distortion of size and shape than with the icosohedral method within each face type.

Map-Projections

The map-projection approaches, on the other hand, use the latitude-longitude grid as a starting point. Mark and Lauzon [1985] proposed a system based upon the Universal Transverse Mercator (UTM) which is used by most military agencies around the world. They begin by dividing the 60 UTM zones into north and south subzones. Each subzone is then subdivided into square patches within which they define a 256 X 256 array of cells. This method coexists nicely with present maps, however the boundaries between zones introduce slight unconformities.

Tobler and Chen [1986] proposed a Lambert cylindrical equal-area projection. This method retains latitude-longitude ideas to create equal area cells. Unfortunately, the variation in shape is tremendous from nearly square at the equator to long, thin spherical rectangles near the poles.

Brown [1993a] introduced the stratified spherical sampling plan (SSSP) which uses a latitude-longitude structure and creates nearly equal area rectangles throughout the sphere. This method does not have the distortion of the Tobler and Chen method. Here, the sphere is cut into "wafers" that are cut parallel to the equator (such as the area between the 70 and 80 degree latitudinal lines on a globe). Upon each wafer a specific latitude-longitude grid is constructed to create almost equal area pieces where distance (horizontal and vertical) is asymptotically preserved within and across wafers.

Each SSSP is made up of 5 parts: the northern cap $C_N(r)$, the southern cap $C_S(r)$, the northern hemisphere $H_N(r)$, the southern hemisphere $H_S(r)$, and the equatorial region $E(r)$. The northern and southern caps and the equatorial region are used as little blocks and separate the two hemispheres that drive the distribution theory.

They can be explicitly calculated by using functions $\gamma_1^*(r), \phi(r), \theta_w(r)$, and integer sequences J_r and v_r, where $\theta_w(r)$ and $\phi(r)$ are the horizontal and vertical generating angles of the latitude-longitude grid on wafer w; J_r, and v_r are the number of $\phi(r)$ vertical angular increments in each wafer and equatorial region, respectively, and $\gamma_1^*(r)$ is used to calculate the top of the first wafer. From these quantities, we can calculate W_r, the number of wafers that the sphere is partitioned into, $n_{w,r}$, the number of $\theta_w(r)$ angles that go around wafer w, and $\gamma_w(r)$, the vertical angle to the top of wafer w.

Denote a point P on a sphere of radius r by its spherical coordinates $P = (r, \theta, \phi)$ where θ is the angle between the positive x-axis and the ray from the origin to P^*, the projection of P onto the xy-plane, and ϕ is the angle between the positive z-axis and the ray from the origin to P.

Given functions $\gamma_1^*(r), \phi(r), \theta_w(r)$, and integer sequences J_r and v_r, calculate $W_r, n_{w,r}$, and $\gamma_w(r)$, mathematically, by first calculating

$$U_r = \frac{1}{\phi(r)J_r} \cdot \{\pi - 2\gamma_1^*(r)\} - \frac{v_r}{J_r}$$

and then put $W_r = U_r - 2z_r^*$, where $z_r^* \in [0, 1)$ is chosen so that W_r is an even integer. Then define the vertical angle to the top of the first wafer as $\gamma_1(r) = \gamma_1^*(r) + z_r^* J_r \phi(r)$. For $1 \leq w \leq W_r/2$, define the vertical angle to the top of wafer w and the number of $\theta_w(r)$ angles that go around wafer w as $\gamma_w(r) = \gamma_1(r) + (w-1)J_r\phi(r)$, $n_{w,r} = n_{W_r+1-w,r} = \lfloor 2\pi/\theta_w(r) \rfloor$, and for the equatorial region, $\gamma_E(r) = \gamma_1(r) + W_r J_r \phi(r)/2$ and $n_{E,r} = \lfloor 2\pi/\theta_E(r) \rfloor$.

For the hemispherical and equatorial regions, we sample at the vertices of the wafer-specific, latitude-longitude grid. Since the shape of each cap is topologically different than that of the wafers, we use a modified hexagonal sampling plan (Matérn [1986]), which provides circular symmetry within the cap.

Define

$$\begin{aligned} H_N(r) &= \bigcup_{w=1}^{W_r/2} \bigcup_{j=0}^{J_r-1} \bigcup_{i=0}^{n_{w,r}-1} P_{i,j}^{w}(r) \text{ and} \\ H_S(r) &= \bigcup_{w=1}^{W_r/2} \bigcup_{j=0}^{J_r-1} \bigcup_{i=0}^{n_{w,r}-1} P_{i,j}^{W_r+1-w}(r) \end{aligned}$$

where in this range for w, $P_{i,j}^{w}(r) = (r,\ i\theta_w(r),\ \gamma_w(r) + j\phi(r))$ and $P_{i,j}^{W_r+1-w}(r) = (r, i\theta_w(r), \pi-(\gamma_w(r)+j\phi(r)))$. Define

$$E(r) = \bigcup_{j=0}^{v_r-1} \bigcup_{i=0}^{n_{E,r}-1} P_{i,j}^{E}(r)$$

where $P_{i,j}^{E}(r) = (r,\ i\theta_E(r),\ \gamma_E(r) + j\phi(r))$. Define

$$\begin{aligned} C_N(r) &= (r,0,0) \cup \left(\bigcup_{j=1}^{\lfloor \gamma_1(r)/\phi(r) \rfloor - 1} \bigcup_{i=1}^{6j-1} P_{i,j}^{N}(r) \right) \text{ and} \\ C_S(r) &= (r,0,\pi) \cup \left(\bigcup_{j=1}^{\lfloor \gamma_1(r)/\phi(r) \rfloor - 1} \bigcup_{i=1}^{6j-1} P_{i,j}^{S}(r) \right) \end{aligned}$$

where $P_{i,j}^{N}(r) = (r,\ i\pi/(3j),\ j\phi(r))$ and $P_{i,j}^{S}(r) = (r,\ i\pi/(3j),\ \pi - j\phi(r))$. A SSSP can now be given by $\mathcal{P}_r = C_N(r) \cup H_N(r) \cup E(r) \cup H_S(r) \cup C_S(r)$.

The SSPSs are the only finite global sampling plans upon which a CLT has been proved (Brown [1993b]). In addition, this is the only global sampling plan upon which a resampling (overlapping bootstrap) algorithm has been designed and strong uniform consistency has been proved for the sample mean (Brown [1993c]).

References

BROWN, J. (1993a), A Sampling Plan for an Isotropic Random Sphere, *submitted.*

BROWN, J. (1993b), A Central Limit Theorem for an Isotropic Random Sphere, *J. Stoch. Geom. and Statis. Apps.,* (to appear).

BROWN, J. (1993c), A Bootstrap Algorithm for an Isotropic Random Sphere, *J. Stoch. Geom. and Statis. Apps.,* (to appear).

CARLSTEIN, E. (1986), The use of subseries values for estimating the variance of a general statistic from a stationary sequence, *Ann. Stat.,* **14**:1171–1179.

DUTTON, G.H. (1989), Modeling Locational Uncertainty via Hierarchical Tessellation, *Accuracy of Spatial Databases,* M.F. Goodchild and S. Gopal (eds.). London: Taylor & Francis.

EFRON, B. (1979), Bootstrap methods: Another lok at the Jackknife, *Ann. Stat.,* **7**:1–26.

FEKETE, G. (1990), Sphere Quadtrees: A New Data Structure to Support the Visualization of Spherically Distributed Data, *Proceedings of the SPIE/SPSE Symposium on Electronic Imaging Science and Technology.*

GASSON, P.C. (1983), *Geometry of Spatial Forms,* Chichester, England: Ellis Horwood Limited.

GOODCHILD, M.F., AND SHIREN, Y. (1989), A Hierarchical Spatial Data Structure for Global Geographic Information Systems, Technical Paper 89-5, National Center for Geographic Information and Analysis, University of California, Santa Barbara.

HARDIN, R.H., AND SLOANE, N.J.A. (1993), A New Approach to the Construction of Optimal Designs, *J. Stat. Plann. and Inference,* **37**:339–369.

HOOKE, R. AND JEEVES, T. (1961), 'Direct Search' Solution of Numerical and Statistical Problems, *J. Assoc. Comp. Mach.,* **8**:212–229.

KÜNSCH, H. (1989), The jackknife and the bootstrap for general stationary observations, *Ann. Stat.,* **17**:1217–1241.

LEONENKO, L. AND YADRENKO, M. (1979), Limit theorems for homogeneous and isotropic random fields, *T. Prob. Math. Stat.,* **21**:113–125, Translated by Szucs, J.

MATÉRN, B. (1986), *Spatial Variation,* Lecture Notes in Statistics, Berlin: Springer-Verlag.

MARK, D. AND LAUZON, J. (1985), Approaches for a Quadtree-Based Geographic Information System at Continental or Global Scales, *Proceedings, Auto-Carto,* **7**:355–364.

POPKO, E. (1968), *Geodesics,* Detroit: University of Detroit Press.

SAMET, H. (1984), The Quadtree and Related Hierarchical Data Structures, *Assoc. Comp. Mach. Computing Surveys,* **16**:187–260.

TOBLER, W., AND CHEN, Z-T. (1986), A Quadtree for Global Information Storage, *Geog. Analysis,* **18**:360–371.

WHITE, D., KIMERLING, A.J., AND OVERTON, W.S. (1992), Cartographic and Geometric Compnents of a Global Sampling Design for Environmental Monitoring, *Cart. and GIS,* **19**:5–22.

WICKMAN, R. E., ELVERS, E. AND EDVARSON, K. (1974), A System of Domains for Global Sampling Problems, *Geografisker Annaler, Series A,* **56**:201–212.

Incorporating Segmentation Boundaries into the Calculation of Fractal Dimension Features

C.E. Priebe[1,2], E.G. Julin[1,4], G.W. Rogers[1,2], D.M. Healy[3], J. Lu[3], J.L. Solka[1], & D.J. Marchette[1]

[1]Systems Research & Technology Department, Advanced Computation Technology, B10
Naval Surface Warfare Center, Dahlgren, VA 22448-5000
Phone: (703) 663-7650, FAX: (703) 663-7999, Email: <cpriebe@relay.nswc.navy.mil>

[2]Center for Computational Statistics, George Mason University, Fairfax, VA 22030

[3]Department of Mathematics & Computer Science, Dartmouth College, Hanover, NH 03755

[4]Department of Mechanical Engineering, University of Houston, Houston, TX 77204

Abstract

The covering method algorithm can be used to calculate power-law feature vectors based on the local texture in an image. These features can then be used for distinguishing between different types of textures. We present a new method of calculating local fractal-based features in the presence of a continuous-valued, irregular and/or incomplete segmentation by use of a Dijkstra potential map. This method produces more accurate power-law features for pixels near a segmentation boundary by altering the size and shape of the local neighborhood in which the calculations take place, thereby producing a more texturally pure neighborhood. This leads to improved texture discrimination since the contribution of multiple textures to the calculation of a given feature vector is reduced or eliminated.

1. Introduction

To oversimplify, those who have studied the utility of using fractal dimension for discriminant analysis in, say, biological images can be grouped into one of two categories. There are those who feel the information inherent in the fractal dimension of a texture should be useful for distinguishing certain classes of tissue even though few conclusive studies have yet been presented, and those for whom the results obtained thus far are unconvincing enough to warrant a decision to move on to other approaches. The optimists feel a system utilizing fractal dimension in conjunction with other information and techniques will be superior to a system which fails to utilize any type of textural information. This paper presents one reason why the results obtained thus far are less impressive than some have expected, introduces a new methodology for extracting fractal dimesion features which circumvents this cause, and indicates, finally, that this modified approach to fractal dimension does indeed live up to the potential for which the optimists' have long held out.

Section 2 presents a description of a modification of the covering method algorithm for estimating fractal dimension which incorporates segmentation boundaries. A qualitative comparison of the procedure with the standard covering method is presented in Section 3. Probability density estimates for the extracted feature vectors are developed and compared. Examples are presented for a standard texture benchmark and for tumor detection in X-ray mammography. It is shown that there is significantly more discriminatory information in the texture features when they are extracted via the new method.

2. Approach

Richardson's power law (Mandelbrot, 1977) provides a functional relation between a measured property of a fractal and a measurement scale. The function is given by

$$M(\varepsilon) = K\varepsilon^{(d-D)}, \qquad (1)$$

where $M(\varepsilon)$ is a measured property of a fractal at scale ε, K is a constant of proportionality, d and D are the topological and fractal dimensions, respectively. Taking the logarithm of Eq. (1) provides the slope and y-intercept of a best-fit line through $\log(M(\varepsilon_i))$ for a set of scales $\{\varepsilon_i\}$ as a set of power law features.

The property $M(\varepsilon)$ we wish to measure is the surface area of the image about a pixel and can be estimated using the covering method (Peleg, et al., 1984). The covering method typically consists of three steps: recursive application of dilation and erosion operators to calculate upper, U, and lower, L, bounding surfaces for scales $\varepsilon_0, \ldots, \varepsilon_{max}$; calculation of an averaged surface area, A, at each scale from U and L; and calculation of power law features from A. When two or more textures are present in an image the morphological operators and the averaging process will *both* lead to erroneous estimates of A and thus the derived features if boundaries between textures are not accounted for. To remedy the errors due to the dilation and errosion operators we utilize the modified dilation and erosion operators (Rogers, et al. 1993, and Julin et al. 1994)

$$U_{i,j}^{\varepsilon+1} = max\left\{\begin{matrix} U_{i,j}^{\varepsilon}+1, \\ [U_{i+1,j}^{\varepsilon}S_{i+1,j}+U_{i,j}^{\varepsilon}(1-S_{i+1,j})], \\ [U_{i-1,j}^{\varepsilon}S_{i-1,j}+U_{i,j}^{\varepsilon}(1-S_{i-1,j})], \\ [U_{i,j+1}^{\varepsilon}S_{i,j+1}+U_{i,j}^{\varepsilon}(1-S_{i,j+1})], \\ [U_{i,j-1}^{\varepsilon}S_{i,j-1}+U_{i,j}^{\varepsilon}(1-S_{i,j-1})] \end{matrix}\right\} ,\tag{2a}$$

and

$$L_{i,j}^{\varepsilon+1} = min\left\{\begin{matrix} L_{i,j}^{\varepsilon}-1, \\ [L_{i+1,j}^{\varepsilon}S_{i+1,j}+L_{i,j}^{\varepsilon}(1-S_{i+1,j})], \\ [L_{i-1,j}^{\varepsilon}S_{i-1,j}+L_{i,j}^{\varepsilon}(1-S_{i-1,j})], \\ [L_{i,j+1}^{\varepsilon}S_{i,j+1}+L_{i,j}^{\varepsilon}(1-S_{i,j+1})], \\ [L_{i,j-1}^{\varepsilon}S_{i,j-1}+L_{i,j}^{\varepsilon}(1-S_{i,j-1})] \end{matrix}\right\} ,\tag{2b}$$

where U^{ε} is the upper surface L^{ε} the lower surface at scale ε and *i,j* are the row and column indices respectively. *S* is a continously valued segmentation map with $S \in [0,1]$, where $S = 0$ for the strongest possible segentation boundary and $S = 1$ for no boundary.

The upper and lower surfaces at scale zero are given by

$$U_{i,j}^{0} = L_{i,j}^{0} = G_{i,j}, \tag{3}$$

where $G_{i,j}$ is the original gray scale image.

It is customary to utilize the average area formula of Peli (Peli, 1990),

$$A_{i,j}^{\varepsilon} = \sum_{k,l \in W_{i,j}} a_{i,j}^{\varepsilon}, \tag{4}$$

where

$$a_{i,j}^{\varepsilon} = \frac{U_{i,j}^{\varepsilon} - L_{i,j}^{\varepsilon}}{2\varepsilon} \tag{5}$$

to reduce the variation of the area from pixel to pixel. In this method the averaging window $W = W(\varepsilon)$ such that at scale *m* the window about *i,j* should be larger than $(2m+1) \times (2m+1)$ so that the window contains sufficient uncorrelated values. However, when the window encompasses multiple textures the averaging process is a source of error.

To reduce or eliminate the effects of averaging multiple textures we introduce a boundary observing adaptive kernel based on Dijkstra potentials (Dijkstra, 1959). In this approach a potential is calculated about every pixel in the image from costs defined below. The potential is then utilized in constructing a kernel for computing the average area about each pixel.

In the current calculations two types of costs are considered. The first is the cost based on the shortest possible path from the current pixel to the window's central pixel. The distance used for the current calculations is based not on the (physical) distance between pixel centers, but rather on the number of steps required to move from the current pixel to the central pixel. This cost is dependent upon the type of connections we allow between pixels. For example, the cost of connecting pixel *k+1,l+1* to *k,l* would be 2 if we constrain connections to the north, east, west, south four nearest neighbors (first we must move to *k,l+1*, or *k+1,l*, then to *k,l*).

The second type of cost is that of being coincident or adjacent to a boundary pixel. For a binary boundary (*S*=0 or 1 only) this cost is set to an arbitrarily large value. If a pixel is not adjacent to a boundary pixel this cost is zero. For continuously valued segmentaion boundaries we utilize the cost function

$$C_{i,j} = \alpha(1 - min(S_{i,j}, S_{i',j'})), \tag{6}$$

where the prime denotes pixels within the neighborhood and α is a parameter describing the amount of information allowed to cross the boundary. Other types of costs or cost functions are easily implemented.

Once the costs have been computed the four nearest-neighbor recursive potential update equation,

$$V_{k,l}^{\alpha+1} = min\left\{\begin{matrix} V_{k,l}^{\alpha}, & \\ V_{k-1,l}^{\alpha}+C_{k-1,l}, & V_{k+1,j}^{\alpha}+C_{k+1,l}, \\ V^{\alpha}{}_{k,l-1}+C_{k,l-1}, & V_{k,l+1}^{\alpha}+C_{k,l+1} \end{matrix}\right\}, \quad \forall\ k,l \in W_{i,j} \tag{7}$$

is iterated to convergence. Here $V_{k,l}^{\alpha}$is the potential at step α and $C_{k,l}$ the sum of costs at pixel *k,l* with

$$V_{k,l}^{0} = \begin{cases} 0 & if\ k,l = i,j \\ \infty & otherwise \end{cases} \tag{8}$$

In the present study we have utilized a window of fixed "radius," *r*, *i.e.* the window is of size $(2r+1) \times (2r+1)$, as opposed to the variable window of Peli. We feel that this is appropriate as long as $r \geq \varepsilon_{max}$. We note that it will be possible for the kernel to be smaller than the window in the vicinity of a boundary.

We may now utilize the Dijkstra potentials in the calculation of the area about pixel *i,j* by performing a weighted summation over the window using

$$A^{\varepsilon}_{i,j} = \frac{\sum_{k,l \in W_{i,j}} a^{\varepsilon}_{k,l} w_{k,l}}{\sum_{k,l \in W_{i,j}} w_{k,l}}, \tag{9}$$

where $a_{k,l}$ is the area calculated by Eqn. (5) at pixel *k,l* and *w* is a weight function based on the Dijkstra potential given by, say,

$$w_{k,l} = \begin{cases} 1 & V_{k,l} < \lambda \\ 0 & otherwise, \end{cases} \tag{10}$$

for a square kernel where λ is a parameter. In Section 3 below we use $\varepsilon_{max} = 5$, $r = 8$, and $\alpha = \lambda = 16$.

3. Results

In this section we present the results of using the above technique on two illustrative examples. The first consists of considering the estimate of the y-intercept value from two Brodatz texture patches (Brodatz, 1966). The ability to obtain a good estimate in the region of transition between the two textures yields superior performance in a change point detection scenario. The second example presented considers an x-ray mammogram and investigates the ability to distinguish a tumorous region from the healthy tissue. Here we consider the estimate of the fractal dimension itself. In both examples the incorporation of boundary information into the calculation of our features is vital to obtaining an acceptable level of performance. Probability density functions are developed using the method of adaptive mixtures (Priebe and Marchette, 1993) and utilize the imposed measure methodology (Priebe, et al., 1994).

3.1 Example 1

Given two textures from Brodatz (Fig 1.1) we consider three regions. The leftmost box (box 1) superimposed on the textures in Figure 1.1 is well within the interior of the left texture and can reasonably be considered a region of pure texture 1 (D17 of Brodatz). Similarly, the rightmost box (box 3) is a region of pure texture 2 (D24 of Brodatz). The middle box (box 2) stradles the boundary between the two textures. This border region contains some pixels from texture 1 and some from texture 2, as well as the boundary.

Figure 1.2 shows (as solid lines) the pdfs obtained from the pure textures in boxes 1 and 3, calculated separately. These pdfs for the different textures are well separated when the regions considered are far from the border and hence uniform in texture. The dashed line in Figure 1.2 shows, however, that when we consider a border region (box 2) the errors arising from calculating power law features over a region containing two distinct textures makes it impossible to determine the structure of the region. This pdf does not convey the fact that the region considered contains exactly two distinct textures. The dotted curve in Figure 1.2 indicates the pdf of the border region (box 2) when a priori boundary information ($S = 0$ or 1) is incorporated into the calculation of the power law features. It is obvious from this pdf that the region being considered is simply made up of two subregions with characteristics corresponding to those in boxes 1 and 3. This superior information is easily translated into superior performance in discriminant analysis or change point detection scenarios.

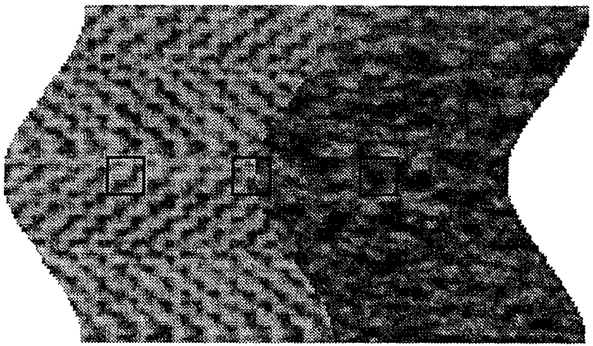


Figure 1.1.
Two adjacent texture patches and the three regions (numbered 1 through 3 from the left) used in Example 1.

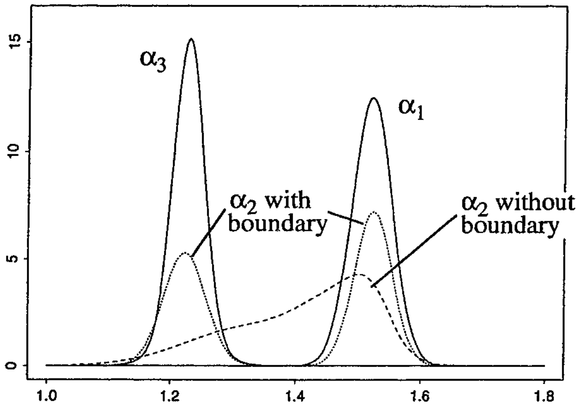


Figure 1.2.
Pdfs of the y-intercept feature for the three regions from Figure 1.1.

3.2 Example 2

For example 2 we consider the mammogram shown in Figure 2.1. We will focus on the boxed region in the upper right. This region contains a tumorous region (biopsy veriwith the radiologist's boundary drawn in. We consider two disjoint regions. The tumorous region (region 1) is the region within the radiologist's boundary. The healthy region (region 2) is the area simultaneously within the box and outside the tumorous region.

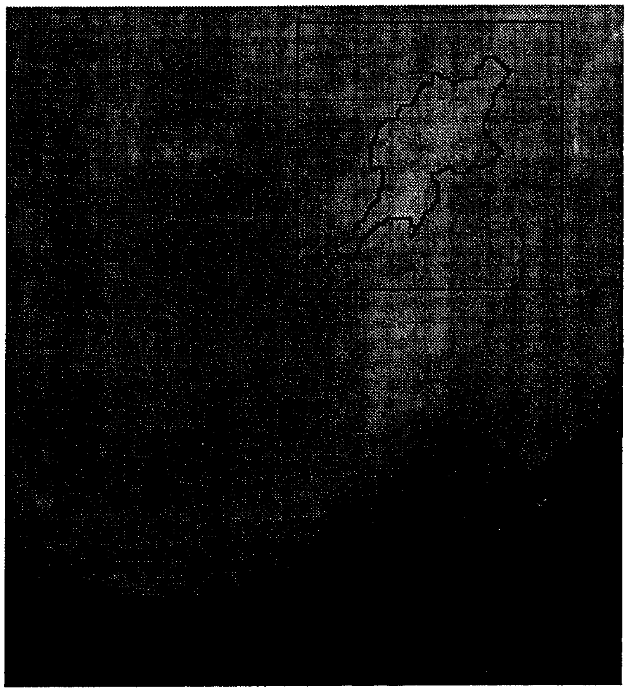


Figure 2.1
Mammogram used in Example 2 with radiologist's boundary of tumorous region overlaid.

Figures 2.2 and 2.3 show, respectively, pdfs for the two regions when the true boundary has been incorporated into the calculation of the features (2.2) and when no boundary is used (2.3). We clearly see that the presence of the boundary in the feature extraction is vital to the utility of the features for distinguishing tumorous tissue from healthy tissue.

Unfortunately, obtaining a true boundary like that shown in Figure 2.1 and used in Figure 2.2 is costly and time consuming. Furthermore, the ultimate utility of this procedure for a real application depends on the ability to automatically generate a boundary that will be useful in this context. Figure 2.4 shows the radiologist's boundary superimposed on a particular wavelet segmentation map. This wavelet map is by no means perfect. The boundary is not closed, it is not necessarily exactly coincident with the radiologist's boundary, it is continuously valued rather than binary, and there is noise. Nevertheless, it generally marks the edge of the tumorous region. When this boundary is used in the feature extraction the resultant pdfs are depicted in Figure 2.5. We see that the separation of the two classes is maintained to a degree similar to that obtained when the radiologist's boundary was employed. Discriminant analysis could be successfully pursued here, as in Figure 2.2, while Figure 2.3 (the no boundary case) leaves little hope.

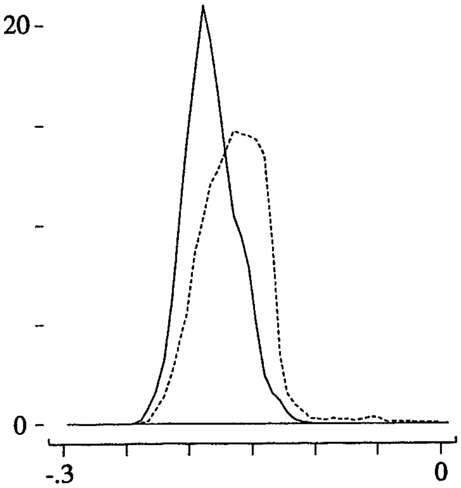


Figure 2.2
Pdfs for fractal dimension from Example 2, calculated using the radiologist's boundary. Solid curve is tumorous tissue, dashed curve is healthy tissue.

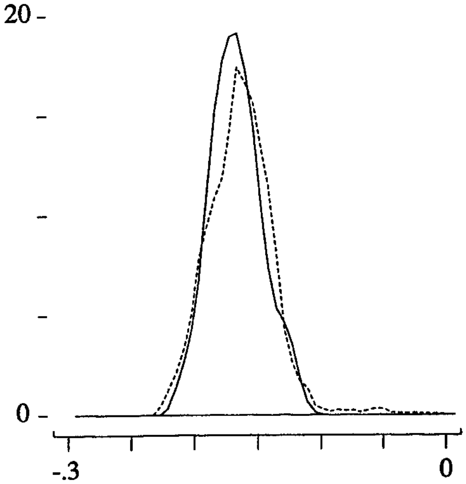


Figure 2.3
Pdfs for fractal dimension from Example 2, calculated with no boundary information. Solid curve is tumorous tissue, dashed curve is healthy tissue.

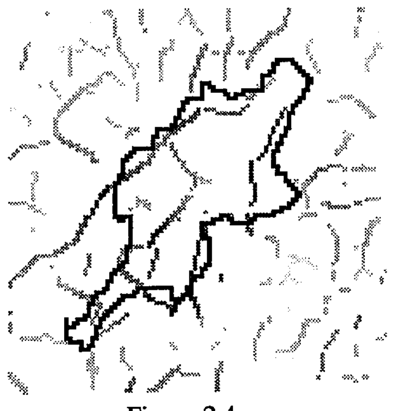


Figure 2.4
Incomplete, grayscale wavelet segementation map with radiologist's boundary overlaid. This continuous valued map is used for Figure 2.5.

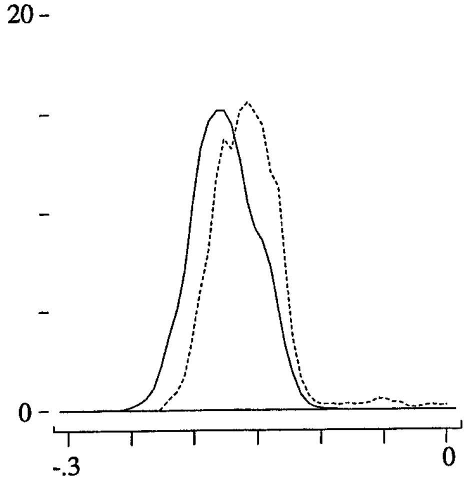


Figure 2.5
Pdfs for fractal dimension from Example 2, calculated using the continuous valued wavelet boundary boundary from Figure 2.4. Solid curve is tumorous tissue, dashed curve is healthy tissue.

4. Discussion

The examples presented in Section 3 indicate that the utility of fractal dimension features for texture discrimination hinges on calculating the features in regions of uniform texture. For applications in which one necessarily must consider border regions between different textures the standard calculations do not provide the necessary capabilities. Incorporating a segemntation boundary into the ´calculation of the texture features, whether it be a true boundary known a priori or a boundary map estimated through a wavelet or other algorithm, greatly improves the discrimination capabilities one can expect.

It is argued that this modificaiton must be considered in any evaluation of the utility of power law features for discriminant analysis, change point detection , or homogeneity analysis whenever texture boundaries come into play.

Acknowledgements

This work was supported in part by the Naval Surface Warfare Center Dahlgren Division Internal Research program. Mammographic image and radiological analysis provided courtesy of the Center for Engineering and Medical Image Analysis at the University of South Florida. The authors gratefully acknowledge Dr. Richard A. Lorey for his careful reading of the paper and many helpful suggestions.

References

Brodatz, P. 1966. *Texture: A Photographic Album for Artists and Designers*. Dover. New York.

Dijkstra, E.W. 1959. "A Note on Two Problems in Connection with Graphs." Numerische Mathematic, **1**, 269-271.

Julin, E.G., G.W. Rogers, C.E. Priebe, and J.L. Solka. 1994. "Calculation of power law features in the presence of segmentation utilizing a Dijkstra potentional based algorithm," *Proceedings of the Conf. on High Performance Computing '94*, pp. 357-362.

Kullback, S. 1959. *Information Theory and Statistics*. Wiley, New York.

Mandelbrot, B.B. 1982. *The Fractal Geometry of Nature*. W. H. Freeman and Company, San Francisco.

Peleg, S., J. Naor, R. Hartley, and D. Avnir. 1984. "Multiple Resolution Texture Analysis and Classification," IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-6**, 518-523.

Peli, T. 1990. "Multiscale fractal theory and object characterization," J. Opt. Soc. Am. A, **7**,1101-1112.

Priebe, C.E. and D.J. Marchette. 1993. "Adaptive Mixture Density Estimation." Pattern Recognition, **26**, no. 5: 771-785.

Priebe, C.E., G.W. Rogers, D.J. Marchette, and J.L. Solka. 1994. "Change Point Analysis with Adaptive Mixture Models," *IGARSS* (to appear).

Rogers, G.W., C.E. Priebe, H. Hayes, and J.L. Solka. 1993. "A Parallel Distributed Processing Algorithm for Power Law Features Which Requires Only Nearest Neighbor Communication." Presented at and to appear in *Proceedings of the SIMTEC/WNN '93*. (San Francisco, CA, Nov 7-10).

Rogers, G.W., C.E. Priebe, and E.G. Julin. 1994. "Calculation and comparison of fractal dimension distributions," *SCS94* (to appear).

Overfitting in Neural Networks

A. B. Owen
Department of Statistics
Stanford University
Stanford, CA 94305
owen@playfair.stanford.edu

Abstract[1]

Feedforward neural networks are widely used as a black box prediction technique. Recent work of Barron (1991) shows that these models are very well suited to approximating structure in high dimensions. This raises the issue of how well they find spurious structure in noise.

This paper presents a diagnostic based on Aldous's Poisson clumping heuristic that describes the extent to which nets can overfit, where in the data such spurious overfitted units are likely to arise and how many local optima the sum of squared error surface (as a function of the network weights) is expected to have.

The diagnostic is simplest for the case of a single hidden unit, but extends in principle to more general problems.

1 Introduction

We consider a nonlinear regression model of the form $Y = \mu(X) + \epsilon$. Here the response variable Y is the sum of a signal $\mu(X)$ and a noise random variable ϵ with mean 0 and variance σ^2. The signal is a function of X, a vector of predictor variables. The form of the signal is

$$\mu(X) = \omega_0 + \sum_{j=1}^{J} \omega_j \phi_j(X, \theta_j) \qquad (1)$$

where the ω_j are scalar parameters ("weights"), the ϕ_j are real valued "activation" functions and the θ_j are vector valued parameters.

The model (1) is an example of an artificial neural network model. This special case is known as a feedforward network with a single hidden layer and a linear output unit. See Hertz, Krogh and Palmer (1991, Chapters 5,6) for an introduction to these models. Commonly used activations are sigmoids such as $\phi(X,\theta) = (1+\exp(-X'\theta))^{-1}$ and Gaussian radial basis functions such as $\phi(X,\theta) = \exp(-\|X-\theta\|^2/2\tau^2)$. In the sigmoid above, X usually includes a component that is always 1 and the corresponding intercept component of θ is known as a "bias". In the radial basis function the parameter τ is a measure of scale that could either be subsumed into θ or held fixed.

2 Asymptotics and Redundant Units

Estimation of model (1) is usually based on training data consisting of n independent observations (X_i, Y_i). Let $\hat\theta_j$ and $\hat\omega_j$ denote estimates of the parameters and $\hat\mu(X)$ denote the resulting estimate of signal.

If model (1) holds, then mild assumptions on the distribution of ϵ and identifiability assumptions on $\omega_j, \phi_j, \theta_j$ produce the usual asymptotics as $n \to \infty$ for $\hat\mu$ estimated by minimizing squared error $\sum_{i=1}^{n}(Y_i - \hat\mu(X_i))^2$. The parameters are estimated consistently (up to some permutations of labels which don't matter) and are asymptotically normally distributed. The mean squared error on the training data is smaller than σ^2, but this optimism is simply accounted for by adjusting for the degrees of freedom used in fitting the model. For details see White (1989).

These asymptotics are suspect for the problem at hand. Partly this is because typical applications use a very large number of parameters. When a large number of parameters are in use, the identifiability assumption becomes questionable. The model (1) is not identifiable, if for example $\omega_J = 0$, for then the corresponding θ_J has no effect on Y. There is thus no "true value" for θ_J and estimates of it have nothing to converge to. This undermines the usual approach to asymptotic theory. The unit $\phi_J(X, \theta_J)$ is said to be redundant, and the corresponding estimate $\phi_J(X, \hat\theta_J)$ is said to be spurious.

It is unlikely in practice that an exactly redundant

[1] This work supported by NSF grants DMS-9011074 and 94-04594

unit will be encountered. But in a model with many units it is reasonable that some of the ω_j will be close to zero and hence that some units are nearly redundant.

Redundant unit asymptotics are like those of broken line regression. Here X is a scalar and

$$\mu(X) = \omega_0 + \omega_1 X + \omega_2 (X - \theta)_+ \qquad (2)$$

where z_+ denotes $\max(z, 0)$, and the only nonlinear parameter is a scalar θ. When $\omega_2 = 0$, the signal is linear in X, and minimizing $\sum_{i=1}^n (Y_i - \hat{\mu}(X_i))^2$ over all broken line regressions reduces squared error by more than it would in a four parameter linear model. The nonlinear parameter θ "uses up" approximately 2 degrees of freedom, according to simulations in Hinkley (1969) and asymptotics in Owen (1991). The maximizing value $\hat{\theta}$ can appear anywhere but it is more likely that spurious bends will appear near the ends of the observed range of X_is.

The questions for neural networks are:

Q1 How many degrees of freedom do the nonlinear parameters in (1) use up?

Q2 Where are the spurious units most likely to appear?

Q3 Which units if any are less prone to overfitting?

3 One Nonlinear Unit

To examine these issues we consider the simplified problem of training a single hidden unit. The model

$$\mu(X) = A(X)\beta + \omega\phi(X, \theta) \qquad (3)$$

has one hidden unit to train, and when $\omega = 0$ that one unit is redundant. The term $A(X)\beta$ is a linear model in some non-adaptive basis functions $A(X)$ with coefficients β. This might be simply a constant, or a linear model in X, or it might include units $\omega_j\phi(X, \theta_j)$ with their nonlinear parameters θ_j frozen at some values and with ω_j subsumed into β. The model without the redundant unit is:

$$\mu(X) = A(X)\beta \qquad (4)$$

Even when (4) is true, the sum of squared errors under (3) will be smaller. For any fixed θ the reduction is

$$S(\theta) = SSE_{(4)} - SSE_{(3)}(\theta) \sim \sigma^2 \chi^2_{(1)}. \qquad (5)$$

The χ^2 result in (5) is exact for normally distributed errors and is an asymptotic approximation otherwise. The reduction of the squared error of model (3) over (4) is

$$S = \sup_{\theta \in \Theta} S(\theta) \qquad (6)$$

and S does not have a χ^2_{1+d} distribution, with $d = \dim(\Theta)$, as one might have expected based on linear model theory.

It is convenient to define a signed root process via

$$Z(\theta) = \pm S(\theta)^{1/2} \sim N(0, \sigma^2) \qquad (7)$$

where the sign of $Z(\theta)$ is the same as that of $\hat{\omega}$ when fitting (3) with θ fixed. Let $Z_{\max} = \sup_{\theta \in \Theta} Z(\theta)$. For large $y > 0$, $P(S > y) \doteq 2P(Z_{\max} > y^{1/2})$.

Suprema of Gaussian random fields, such as $Z(\theta)$, have been well studied. At any θ, for smooth processes, Z and its first two derivatives have a joint Gaussian distribution. A local maximum of Z above Z_0 is a point θ such that $Z(\theta) > Z_0$, the gradient of Z vanishes at θ and the Hessian of Z is negative definite at θ. A standard tail approximation is

$$\begin{aligned} P(Z_{\max} > Z_0\sigma) &\doteq E(\#\text{Local Maxima} > \mathrm{Z}_0\sigma) \\ &\doteq \int_\Theta \lambda(\theta) d\theta \end{aligned}$$

where $\lambda(\theta)$ is the intensity of high local maxima of Z near θ.

For one dimensional intervals Θ of finite length, this formula is the expected number of "upcrossings" of the level $Z_0\sigma$ by the process $Z(\theta)$. If one adds the probability that Z exceeds $Z_0\sigma$ at one end of Θ one gets Rice's formula which is in this case an upper bound on $P(Z_{\max} > Z_0\sigma)$. For stationary fields, this formula reduces to the volume of Θ times an intensity that is constant in θ. See Adler (1981, Chapter 6). The formula above is taken from Aldous (1989, Chapter J7). This formula is the lead term in the more accurate but more difficult formulas obtained by Siegmund and Knowles (1989). The more accurate formulas take more care around the boundary of Θ.

The intensity function is

$$\lambda(\theta) = (2\pi)^{-(d+1)/2} Z_0^{d-1} e^{-Z_0^2/2} |\Lambda(\theta)|^{1/2}$$

where $|\Lambda|$ denotes the determinant of Λ and

$$\Lambda(\theta_0) = -\frac{\partial^2}{\partial\theta\partial\theta'} E(\sigma^{-2} Z(\theta_0) Z(\theta))|_{\theta=\theta_0}$$

is the Hessian of the correlation matrix of $Z(\theta)$ evaluated at θ_0.

Owen (1993a, Theorem 2) gives an expression for the rs entry of $\Lambda(\theta)$. Let Φ be the vector of n values $\phi(X_i, \theta)$, let Φ_r be the vector of $\partial\phi(X_i, \theta)/\partial\theta_r$, and let M be the projection matrix on the space spanned by the matrix with n rows given by $A(X_i)$. Define the inner product

$< g,h >= g'(I-M)h$ and define $\gamma_r =< \Phi_r, \Phi > / < \Phi, \Phi >$. Then

$$\Lambda_{rs} =< \Phi_r - \gamma_r \Phi, \Phi_s - \gamma_s \Phi > / < \Phi, \Phi > . \tag{8}$$

This equation may be better understood as an algorithm: Construct the vectors Φ, Φ_r, Φ_s by evaluating $\phi(X_i, \theta)$ and its gradient with respect to θ. Then replace them by their residuals after fitting linear model on the predictors $A(X)$. Then find the partial correlation of the resulting Φ_r and Φ_s variables after adjusting for Φ.

The result provides $\Lambda_{rs}(\theta)$. Doing this for all r and s and taking the determinant allows one to calculate the intensity $\lambda(\theta)$.

Thus for one nonlinear unit, we have a way to approximately answer the questions raised above:

A1 Integrate λ over Θ and compare with chisquare tail probabilities.

A2 Maximize or plot λ (or $|\Lambda|^{1/2}$) over Θ.

A3 Compare λ (or $|\Lambda|^{1/2}$) for different activations $\phi(X, \theta)$.

In A2 and A3 the use of $|\Lambda|^{1/2}$ is a little simpler since unlike $\lambda(\theta)$, it does not depend on Z_0.

4 Results and Examples

The intensity function $\lambda(\theta)$ can be evaluated either numerically or theoretically. Based on this, one can find predictions of the Poisson clumping heuristic:

P1 Long tailed units ϕ lead to fewer local maxima and use fewer degrees of freedom in noise.

P2 Spurious bent planes are more likely near the convex hull of the X's.

P3 Spurious sigmoidal units are more likely to pass through the middle of the X's.

P4 Spurious radial basis units are more likely when the radius is small.

P5 Those small radius units are likely to be found near voids in the X's.

Since the method works by estimating the expected number of high local maxima, it also sheds some light on which types of units are likely to make global optimization difficult.

Figure 1 shows 216 predictors $X \in R^2$ for a synthetic data set. For a Gaussian radial basis function model $\phi(X, \theta) = \exp(-\|X-\theta\|^2/2\tau^2)$ in (3). With this model form, θ is in the same space as the X_i. Figure 2 shows $|\lambda(\theta)|^{1/2}$ for this model taking $\tau = 0.5$, $A(X) = 1$ and $\sigma^2 = 1$. We use $\sigma^2 = 1$ in all examples in this section.

The peak of $\lambda(\theta)$ is in the middle of the X_i set. There is a second peak between the main body of the data and a small cluster near $(3, 1)$. There are ridges extending away from the data along lines equidistant from pairs of points on the convex hull of the X_i. For smaller τ the function $\lambda(\theta)$ generally increases and the ridges become very high and sharp. The ridges correspond to θ-regions in which small changes in one unit can explain either of two potential outliers, or perhaps both of them, if they have the same sign. For small τ large spikes can appear over the centers of gaps in the point cloud. In these locations small changes in θ can make big changes in what the unit explains.

In order to plot the results for sigmoidal units and other activations which are functions of projections of the data, we turn to polar coordinates. For $\theta = (\theta_1, \theta_2)'$, let

$$\pi(X_i, \theta) = X_{i1}\cos(\theta) + X_{i2}\sin(\theta_1) - \theta_2.$$

so that θ_1 is an angle and θ_2 is a radius. Figure 3 shows $\lambda(\theta)$ for a sigmoidal radial basis unit

$$\phi(X_i, \theta) = (1 + \exp(-\pi(X_i, \theta)/\tau))^{-1}.$$

Here $\tau = 0.5$ and $A(X) = 1$. The points in the plot trace out the convex hull of the data from Figure 1. That is for a list of angles θ_1, the maximum and minimum of $X_{i1}\cos(\theta_1) + X_{i2}\sin(\theta_1)$ over the X_i is plotted. Figure 3 shows that spurious sigmoids are more likely to have their linear regions passing through the center of the data than near the convex hull of the data. Decreasing τ makes the sigmoids approach "threshold" units, and this generally increases $|\Lambda|$. (With threshold units, the process $Z(\theta)$ is not smooth enough to apply Theorem 2 of Owen (1993a), but the Poisson clumping heuristic may be applied in another form.)

Figure 4 shows $|\Lambda(\theta)|^{1/2}$ for crease units of the form $\phi(X_i, \theta) = \pi(X_i, \theta)_+$. Again $A(X) = 1$, but for this activation, the spurious events are much more likely near the convex hull of the predictors. Note that taking $A(X) = (1, X')$ makes models (4) and (3) into a plane and bent plane respectively.

Figure 5 shows $\lambda(\theta)$ for hyperbolic fold units of the form $\phi(X_i, \theta) = \pi(X_i, \theta)/2 + (\tau^2 + \pi(X_i, \theta)^2/4)^{1/2}$. For Figure 5, $\tau = 0.5$. Note that as τ decreases to zero, the hyperbolic folds become bent plane creases.

For Figure 6 a sigmoidal unit is considered with $A(X) = (1, \phi(X, \theta_0))$ where $\theta_0 = (\pi/4, 1)$. That is, a second sigmoidal unit is being trained while the first one is held with it's angle at $\pi/4$ and it's radius at 1.0. The resulting plot of $|\Lambda|^{1/2}$ shows that the second unit being trained has a tendency to be close to the first one being held fixed. This suggests that, if units are trained sequentially, that spurious units might arise near units already included in the model. This behavior arises for radial basis unit and for crease units too. It is somewhat weaker for the hyperbolic folds.

Owen (1993b) makes some simplifying assumptions (large n, X spherical Gaussian in d dimensions) and develops an approximation of the form

$$P(S > y) \simeq \delta^{(d-1)/2} P(\chi^2_{(d)} > y) \qquad (9)$$

for units $\phi(X, \theta)$ with fixed radius $||\theta||$. In this case the multiplier δ depends on the radius and of course on the type of unit. The main conclusion is that short tailed units have larger values of δ. For some long tailed units the resulting δ is close to one, indicating that, for such units, redundancy does not make large changes to the asymptotics. Short tailed sigmoidal units are ones where the distribution function corresponding to the sigmoid used has short tails. For example the Cauchy distribution has very long tails, the uniform distribution function has very short tails and the widely used logistic sigmoids have tail lengths between these extremes. Fold units that approximate creases are defined through the integral of a sigmoidal function. The fold has short or long tails according to whether the sigmoid does.

5 Many Units

It is possible to extend this method to problems with many units, though it is harder to find simple descriptions of the results. Suppose that for $j = 1, \ldots, J$ we have $\theta_j \in \Theta_j$. Let Θ_0 be the unit hemisphere in J dimensions, with a positive J'th component. Let $(\theta_{01}, \ldots, \theta_{0J})'$ be a point in Θ_0. Then we may write (1) as

$$\begin{aligned} \mu(X) &= \omega_0 + \omega \sum_{j=1}^{J} \theta_{0j} \phi(X, \theta_j) && (10) \\ &= \omega_0 + \omega \varphi(X, \vartheta) && (11) \end{aligned}$$

where $\vartheta \in \Theta_0 \times \Theta_1 \times \cdots \times \Theta_J$ subsumes all the nonlinear parameters θ_j and all but one degree of freedom of ω_1 through ω_J and φ is a nonlinear function of X.

Sun (1989) uses this construction in studying p values for projection pursuit regression.

Figure Captions

Figure 1 Shown are 216 points $X_i \in R^2$. These are a synthetic data set of predictors.

Figure 2 The points are those of Figure 1. The contours are those of $|\Lambda^{1/2}|$ for a Gaussian radial basis function with radius $\tau = 0.5$.

Figure 3 The contours are those of $|\Lambda^{1/2}|$, in polar coordinates, for a sigmoidal unit with inverse slope $\tau = 0.5$. The points describe the convex hull of the data set in Figure 1.

Figure 4 The contours are those of $|\Lambda^{1/2}|$, in polar coordinates, for a crease (bent-plane) unit. The points describe the convex hull of the data set in Figure 1.

Figure 5 The contours are those of $|\Lambda^{1/2}|$, in polar coordinates, for a hyperbolic unit with inverse slope $\tau = 0.5$. The points describe the convex hull of the data set in Figure 1.

Figure 6 The contours are those of $|\Lambda^{1/2}|$, in polar coordinates, for a sigmoidal unit with inverse slope $\tau = 0.5$. Another sigmoidal unit, with nonlinear parameter frozen at $\theta = (\pi/4, 1.0)$ is included in the linear portion of the model. The points describe the convex hull of the data set in Figure 1.

References

Adler, R.J. (1981) *The Geometry of Random Fields,* Wiley, New York.

Aldous, D. (1989) *Probability Approximations via the Poisson Clumping Heuristic,* Springer-Verlag, New York

Barron, A.R. (1991) "Approximation and Estimation Bounds for Artificial Neural Networks" *Proc. Fourth Annual Workshop on Computational Learning Theory,* pp. 243–249, Morgan Kauffman, San Mateo CA

Hertz, J., Krogh, A. and Palmer, R.G. (1991) *Introduction to the Theory of Neural Computation,* Addison-Wesley, Redwood City, CA.

Hinkley, D.V. (1969) "Inference About the Intersection in Two Phase Regression", *Biometrika,* Vol. 56, 495–504

Owen, A.B. (1993a) "Poisson Clumping and Redundant Units" Technical Report No. 427, Department of Statistics, Stanford University

Owen, A.B. (1993b) "Redundant Units in High Dimensions" Technical Report No. 432, Department of Statistics, Stanford University

Siegmund, D.O. and Knowles, M. (1989) "On Hotelling's approach to testing for a nonlinear parameter in regression," *I.S.I. Review,* Vol. 57, 205–220

Sun, J. (1989) "P-Values in Projection Pursuit" Dissertation, Department of Statistics, Stanford University

White, H. (1989) "Learning in Artificial Neural Networks: A Statistical Perspective", *Neural Computation,* Vol. 1, pp. 425–464

Figure 1

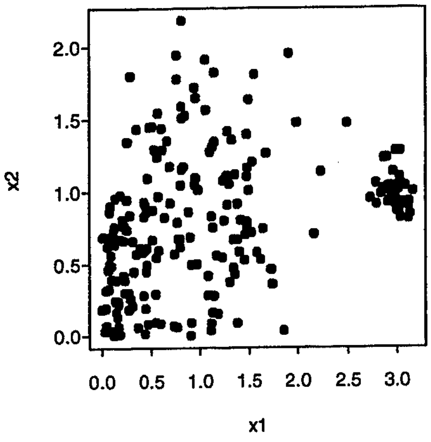


Figure 2

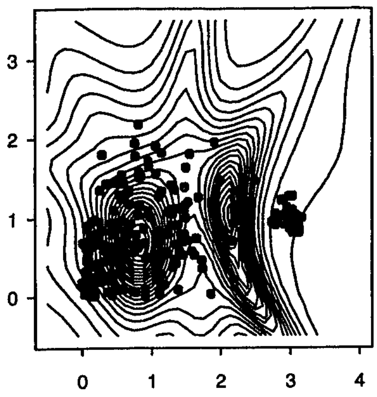


Figure 3

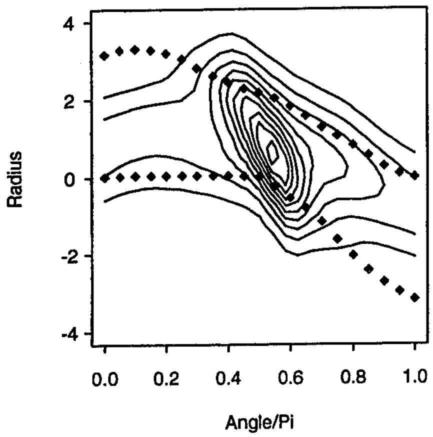


Figure 4

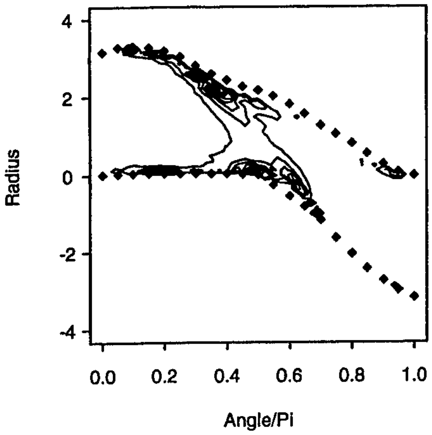


Figure 5

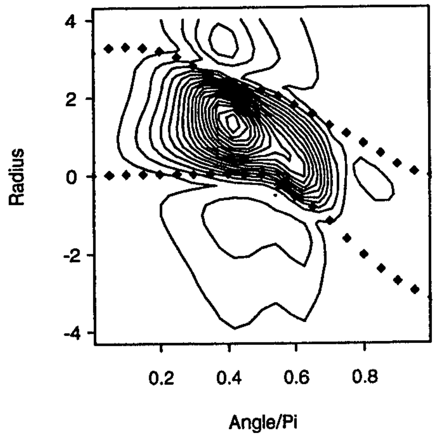


Figure 6

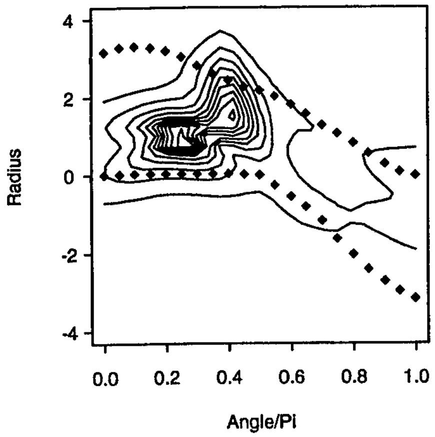


Figure 1

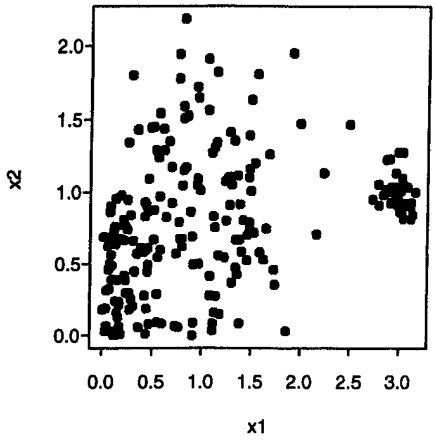


Figure 2

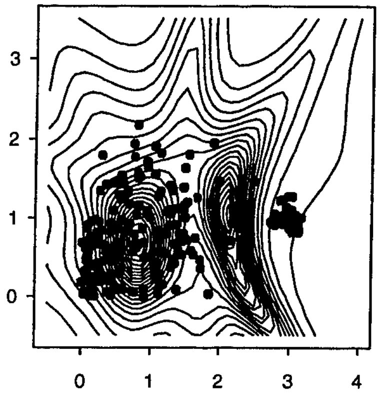


Figure 3

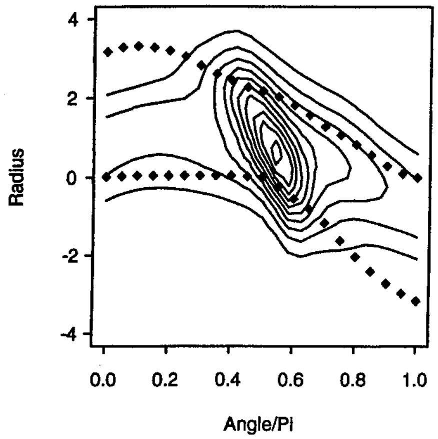


Figure 4

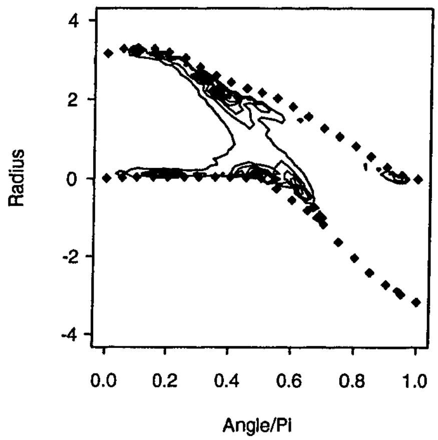


Figure 5

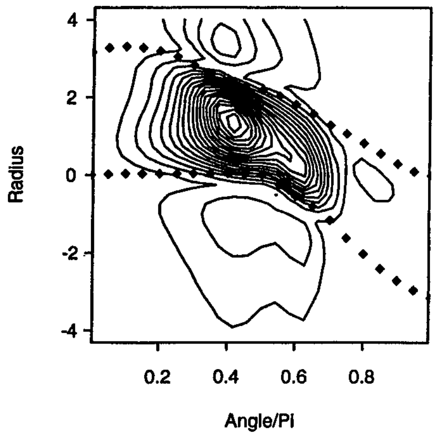
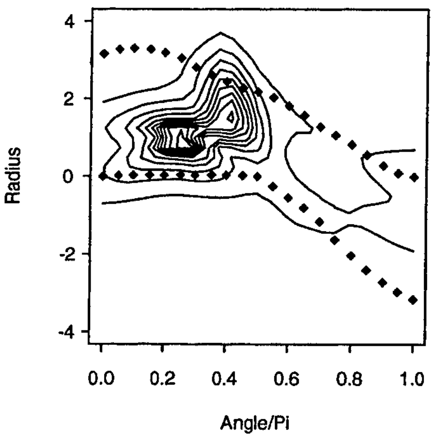


Figure 6



Likelihood Profiles for Studying Non-Identifiability

Christian Ritter
Institut de Statistique
Université Catholique de Louvain,
34, voie du Roman Pays
1348 Louvain-la-Neuve
Belgium

Abstract

The use of likelihood profiles for exploring and measuring non-identifiabiliy and near non-identifiability is discussed. The method is then applied to the estimation of normal-gamma stochastic frontier models used in econometrics. It is shown that these models are practically non-identifiable for samples sizes up to several hundreds of observations.

Keywords: Frontier models

1. Introduction

This paper deals with the following problem frequently encountered in practice. A standard parametric model exists for a certain type of data sets, but the researcher has the impression that the choice of this model is somewhat arbitrary and that a more flexible extension might be more appropriate. The natural move is to add a parameter to increase flexibility and to estimate this parameter together with the quantities one is interested in from the data. Unfortunately, this can easily turn a well-posed problem into a non-identifiable or nearly non-identifiable one. Likelihood profiles can be used to explore such situations.

The tool, profiling, is not new and ample literature exists on various of its aspects. However, except for the work of Bates and Watts (1988) authors have mostly concentrated on the properties of profiles in the context of elimination of nuisance parameters (Barndorff-Nielsen, 1983; Barndorff-Nielsen, 1986) and less on their value for the purpose of exploration (Ritter and Bates, 1993).

The paper begins with an introduction of the notation of the problem and of the terminology of likelihood profiles. In Section 3, a concrete problem, the estimation of normal-exponential and normal-gamma stochastic frontier models (Aigner, Lovell and Schmidt, 1977; Stevenson, 1980; Meeusen and van den Broeck, 1977; Greene, 1990) is described. In Section 4, a strategy for using likelihood profiles to study this problem is laid out. In Section 5, the results of a simulation are reported. The paper is concluded by a discussion of the results.

2. Notations and Terminology

We suppose that data are generated as continuous random variables from a parametric model as

$$X_i \overset{iid}{\sim} F_{\boldsymbol{\theta}}(\mathbf{x}); \quad \boldsymbol{\theta} \in \Theta \subset \mathbf{R}^k, \qquad (2.1)$$

where Θ is a nice connected domain, and where F has density f which is twice continuously differentiable in $\boldsymbol{\theta}$. The corresponding likelihood is denoted by $L(\boldsymbol{\theta}|\mathbf{x}) = f_{\boldsymbol{\theta}}(\mathbf{x})$ and the log-likelihood by $l(\boldsymbol{\theta}|\mathbf{x})$.

Moreover, we assume that inference is conducted by maximum likelihood. That is, for a sample $\mathbf{x} = (x_1, ..., x_n)$ the point-estimate of $\boldsymbol{\theta}$ is obtained by

$$\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}|\mathbf{x}), \qquad (2.2)$$

and confidence regions are computed by either using the inverse information matrix

$$\hat{\boldsymbol{\Sigma}} = \left[D^2_{\boldsymbol{\theta}} \log L(\boldsymbol{\theta}|\mathbf{x}) \Big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} \right]^{-1} \qquad (2.3)$$

or the χ^2 approximation of the log-likelihood.

If we are worried that the model might not be sufficiently flexible, we can try to find an extension by incorporating an additional parameter ψ. We denote the likelihood after adding ψ as $L(\boldsymbol{\theta}, \psi|\mathbf{x})$. Frequently, the original model corresponds to a particular choice of $\psi = \psi_0$ for which $L(\boldsymbol{\theta}, \psi_0|\mathbf{x}) \propto L(\boldsymbol{\theta}|\mathbf{x})$. If $L(\boldsymbol{\theta}, \psi|\mathbf{x})$ is smooth in the joint parameter vector and if ψ_0 is in the interior of the domain of ψ, the usual likelihood-ratio test can be used to check whether the data require the extended model or not.

Frequently, however, maximum likelihood estimates are much harder to find for the extended model than for the original one. The information contained in common finite samples may not suffice to pin down ψ and the estimates of the components of $\boldsymbol{\theta}$ may strongly depend on ψ. In this situation, virtually all precision in the estimation of $\boldsymbol{\theta}$ is lost by going from the original to the extended model. That is, the extended model becomes practically non-identifiable.

In order to assess how adding ψ to the model affects the estimation of $\boldsymbol{\theta}$ we can try to compute the profile trace

$$\tilde{\boldsymbol{\theta}}(\psi) = \text{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \psi|\mathbf{x}) \qquad (2.4)$$

and the profile value of the log-likelihood

$$\tilde{l}(\psi) = \max_{\boldsymbol{\theta}} l(\boldsymbol{\theta}, \psi|\mathbf{x}) = l(\tilde{\boldsymbol{\theta}}(\psi), \psi|\mathbf{x}). \qquad (2.5)$$

If a joint maximum likelihood estimate $(\hat{\boldsymbol{\theta}}, \hat{\psi})$, exists, we can carry this out by re-maximizing the likelihood for discrete values of ψ starting at $\hat{\psi}$ and gradually moving outward. This assures that good starting values for the re-maximization are always available. Alternatively, if no joint maximum can be found but if the original model is a special case of the extended model at ψ_0 one can start with the estimates of the original model and move gradually away from ψ_0. If no obvious starting point is available, a grid of ψ values has to be laid out and the conditional optimizations have to be attempted directly. Once the profile trace and the profile values have been computed for a sufficiently far reaching and fine grid of ψ values, intermediate values can be obtained by spline interpolation. The existence of profiles can only be guaranteed under severe regularity conditions and the reader should keep in mind that computing profiles is an exploratory technique which will work in many but not all situations.

3. A Stochastic Frontier Model

A typical case where practical non-identifiability is observed is the transition from a normal-exponential to a normal-gamma stochastic frontier model for econometric data. Such frontier model have the structure

$$Y_i = \mu + x_i'\beta - z_i + \nu_i, \qquad (3.1)$$

where Y_i represents the observed output (passenger miles for airlines, for example) and $\mu + \mathbf{x}_i'\boldsymbol{\beta}$ the optimal output which can be obtained from the vector of inputs $\mathbf{x}_i = (x_{i;1}, ..., x_{i;p})$ (which could be labor, capital, fuel, etc.). The parameters μ and $\boldsymbol{\beta}$ are unknown and have to be estimated from the data. The two error terms z_i and ν_i represent the inefficiency of unit i and the measurement error. The component z_i is restricted to be positive, while ν_i is usually treated as normally distributed with an unknown variance σ^2. There are several choices for a distribution of the z_i. Good estimation properties can be obtained using an exponential or a half-normal distribution. The disadvantage of these choices are that the shape of the distribution of the inefficiencies is imposed without scientific reason. On can avoid such a hard choice by using a gamma distribution for the inefficiencies instead (Greene, 1990). Gamma distributions are very flexible and contain the exponential distribution as a special case when the shape parameter is equal to one. Unfortunately, maximum likelihood estimation is much more difficult for the normal-gamma model than for the normal-exponential model.

In the following discussion, we denote the variance of the normally distributed noise component ν_i by σ^2, the scale parameter of the exponential or gamma inefficiencies by λ, and the shape parameter of the gamma distribution by α. We assume that the z_i and the ν_i are all independent.

4. Analyzing the Normal-Gamma Model by Likelihood Profiles

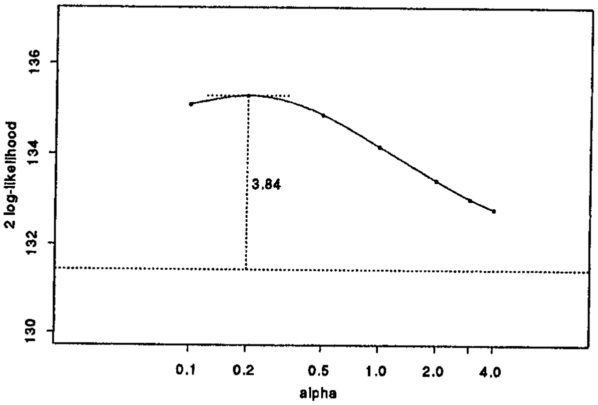


Figure 1: Profile values for the normal-gamma model of the American Electric Utilities (Greene, 1990)]. The evaluated points are joined by an interpolating spline.

The transition from the normal-exponential to the normal-gamma stochastic frontier model is a show-case for the use of likelihood profiles. Suppose that the likelihood of the normal-gamma model has been optimized for fixed values $\alpha_1 < \alpha_2 < \cdots < \alpha_p$ covering a range from distributions more extreme than the exponential (i.e., $\alpha < 1$) to distributions close to normal (i.e., $\alpha >> 1$) and that the corresponding profile trace and the profile values of the log-likelihood are $\tilde{\boldsymbol{\theta}}_1, \tilde{\boldsymbol{\theta}}_2, ..., \tilde{\boldsymbol{\theta}}_p$ and $\tilde{l}_1, \tilde{l}_2, ..., \tilde{l}_p$ (here $\boldsymbol{\theta}$ denotes the combined parameter vector $(\mu, \boldsymbol{\beta}', \sigma^2, \lambda)$). Suppose also that the joint maximum likelihood estimate $(\hat{\boldsymbol{\theta}}, \hat{\alpha})$ was found and is among those values. By the χ^2 approximation of the likelihood ratio statistic we obtain

$$2\left[l(\hat{\boldsymbol{\theta}}, \hat{\alpha}) - l(\hat{\boldsymbol{\theta}}, \alpha)\right] \approx \chi_1^2. \qquad (4.1)$$

This enables us to define likelihood intervals $I_{\{1-\omega\}}$ for α with approximate $1-\omega$ coverage by

$$I_{\{1-\omega\}} = \left\{ \alpha \,\middle|\, \tilde{l}(\alpha) > l(\hat{\boldsymbol{\theta}}, \hat{\alpha}) - \frac{1}{2}\chi_1^2(1-\omega) \right\}. \quad (4.2)$$

In practice, for a coverage probability of 95%, we can plot the profile values $2\tilde{l}_i$ versus the α_i and draw a line $\chi_1(.95)^2 \approx 3.84$ below the observed maximum $2\hat{l}$. The range of α values corresponding to points above the

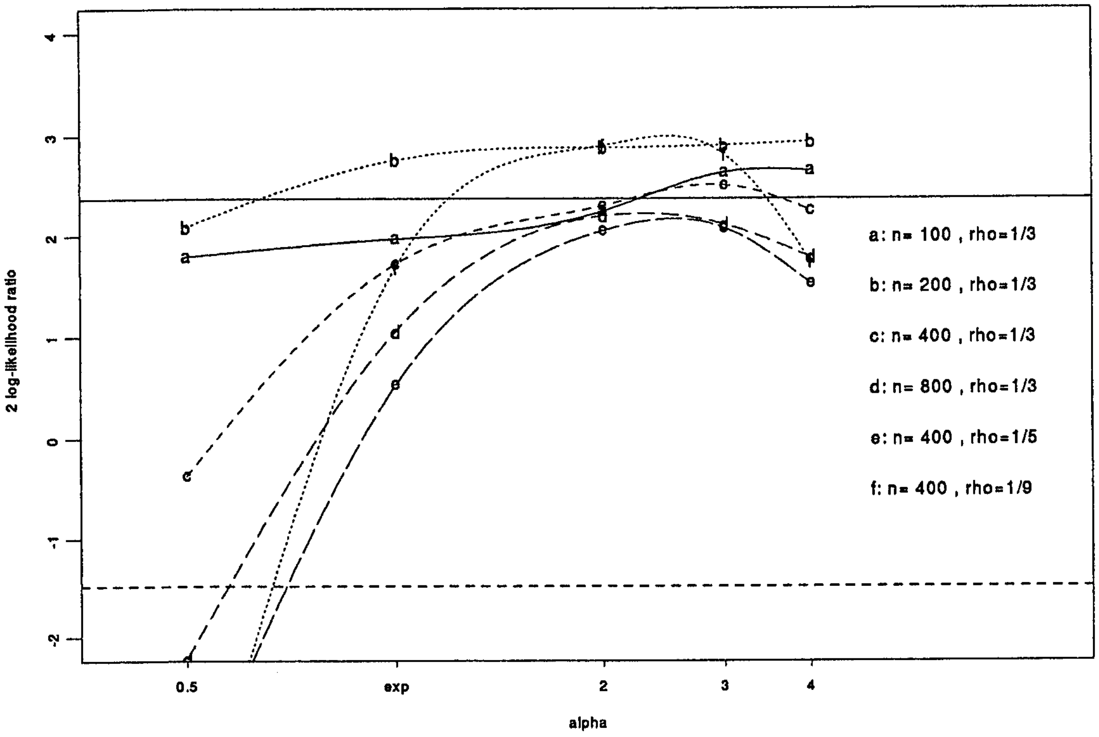


Figure 2: Medians of $2(\tilde{l}_j(\alpha_i) - l_{0;j})$ for each combination of n and ρ. The abscissa is on a logarithmic scale and the evaluated points are joined by interpolating splines.

line provides an approximate confidence interval for α and also a simple graphical means for judging whether α is well-determined by the data. Figure 1 shows such a plot for a normal-gamma model of the efficiencies of American Electrical Utilities analyzed by Christensen and Greene (1976) and Greene (1990).

We see that $2\tilde{l}$ exceeds considerably the lower line for all chosen values of α. None of those values is therefore rejected by the likelihood ratio test. This indicates that the data (123 records) do not contain sufficient information to tie down α. Ritter and Simar (1993) show that the imprecision in the estimation of α carries over to the quantities of econometric interest.

5. Simulation of Special Cases

In this section, we use simulations from a specific but typical normal-gamma model to show how the sample size and the share of the total variance attributed to the noise component ν_i affect the estimation properties of α.

The special case considered here is the normal-gamma model

$$Y_i = \mu - z_i + \nu_i \tag{5.1}$$

with frontier $\mu = 0$ and shape parameter $\alpha = 2$. The choice of the shape parameter corresponds to a distribution which is clearly not exponential, but still far from normal. The parameters characterizing the estimation properties are the sample size n and the ratio $\rho = \sigma^2/(\alpha\lambda^2 + \sigma^2)$, the proportion of "noise" in the total variance. For example, the choice $\rho = 1/3$ implies that 1/3 of the total variability comes from the noise component and 2/3 from the inefficiencies. An allocation of 1/5 to 1/2 of the total variance to the noise component is typical and has for example been observed with the the American Electric Utility data.

For any choice of n and ρ data sets can be simulated. These data sets can then be analyzed by maximum likelihood and, in particular, their profile traces and values can be computed with respect to α.

Recall that the true parameters are known and thus provide the likelihood $l_0 = l(\theta, \alpha|\mathbf{x})$. For fixed $\alpha = 2$, the profile value $\tilde{l}_2 = \tilde{l}(2)$ relates to l_0 via the approximation $2(\tilde{l}_2 - l_0) \approx \chi^2_3$ with a χ^2 distribution with three degrees of freedom. For each simulated data set, the true likelihood l_0 can be computed and used to offset the profile values $\tilde{l}(\alpha_i)$ thus allowing comparisons of the profile values across data sets.

For example, $r_{i,j} = 2(\tilde{l}_j(\alpha_i) - l_{0;j})$ can be computed for simulated data sets $j = 1, ..., m$ and summaries, such as medians and quartiles can be retained for each position α_i. A graphical superposition of the medians for each of the settings above can provide a convenient assessment of the estimation properties of α.

6. Results of The Simulation Study

	$\rho = \sigma^2/(\sigma^2 + \alpha\lambda^2)$			
n	1/9	1/5	1/3	1/2
100			x	x
200			x	
400	x	x	x	
800			x	

Table 1: Scenarios for simulation.

For each scenario indicated in Table 1, 60 data sets were simulated for each of them the profile of the log-likelihood with respect to α was computed. Figure 2 shows the medians of $2(\tilde{l}_j(\alpha_i) - l_{0;j})$ for each scenario. The medians for the same scenario are joined by smooth curves. The solid line represents the expected value of the median of a χ_3^2 distribution with three degrees of freedom; the dashed line $\chi_1^2(0.95) = 3.84$ units below denotes the cutoff corresponding to a 95% confidence likelihood region. As we expect the observed medians for $\alpha = 2$ are close to the theoretical median of the χ_3^2 distribution. Moreover, all points except for $\alpha = 0.5$ of the cases $(800, 1/3)$, $(400, 1/9)$, and $(400, 1/5)$ lie above the dashed line. This suggests that the estimation of α is very poor when the sample size is small and when there is a considerable amount of noise.

7. Discussion

Profiles can provide convenient tools for exploring likelihoods in situations of near non-identifiability. The results can be displayed using simple graphics and are easy to interpret. In the context of normal-gamma stochastic frontier models, this approach yielded the insight that in general large sample sizes are needed to estimate α well. Sample sizes of 100 or 200 observations, which are common in practice, are clearly insufficient.

Gradually, profiling algorithms are finding their way into standard statistical software packages. Explicit profiling algorithms are already available in S and Splus. In other packages, profiling is an implicit ingredient in procedures for Bayesian inference. This is the case in Xlispstat, where the procedure for computing the Laplacian approximation of a marginal relies on the computation of a profile.

References

Aigner, D., Lovell, C. and Schmidt, P. (1977). Formulation and estimation of stochastic frontier production function models, *Journal of Econometrics* **6**: 21–37.

Barndorff-Nielsen, O. (1983). On a formula for the distribution of the maximum likelihood estimator, *Biometrika* **70**: 343–365.

Barndorff-Nielsen, O. (1986). Inference on full or partial parameters based on the standardized signed log likelihood ratio, *Biometrika* **73**: 307–322.

Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis and Its Applications*, John Wiley, New York.

Christensen, L. and Greene, R. (1976). Economics of scale in US electric power generation, *Journal of Political Economy* **84**: 653–667.

Greene, W. (1990). A gamma-distributed stochastic frontier model, *Journal of Econometrics* **46**: 141–163.

Meeusen, W. and van den Broeck, J. (1977). Efficiency estimation from Cobb-Douglas production functions with composed error, *International Economic Review* **8**: 435–444.

Ritter, C. and Bates, D. (1993). Profile methods, *Technical Report 93-31*, Institut de Statistique, Universié Catholique de Louvain, B-1348, Louvain-la-Neuve, Belgium.

Ritter, C. and Simar, L. (1993). Pitfalls of the normal-gamma stochastic frontier model, manuscript available from the authors at the Institut de Statistique, Université Catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium.

Stevenson, R. (1980). Likelihood functions for generalized stochastic frontier estimation, *Journal of Econometrics* **13**: 57–66.

Statistical fit of financial models: tools, workbenches and environments

A.F. Gualtierotti
HEC and IDHEAP, University of Lausanne
CH-1015 Lausanne, Switzerland

ABSTRACT

The evaluation of statistical procedures in the area of finance requires powerful and rich computer environments. Requirements for such environments are stated and their need illustrated with the example of geometric Brownian motion.

1 Introduction

Computer intensive methods and the interface between statistics and computing seem to carry nowdays a specific and rather restrictive meaning, that of single statistical techniques or methods which rely heavily on the computer for implementation. Thus LMS (Least Median of Squares) regression [13] requires that many systems of linear equations be solved. Many problems in statistics have their source outside of it and require that broad arrays of mathematical, statistical and numerical techniques be put to bear on sizeable areas of a particular discipline or set of such. The discipline considered here for illustration is that area of finance which deals with contingent claims [4] and, in it, the simplest model, geometric Brownian motion (GBM henceforth), shall be chosen. The computer intensive aspect of this problem area is due to two basic factors. The first is the high number of mathematical, statistical and programming techniques that must be marshalled to progress towards a solution. The second is the limitation inherent in all analytical developments when numerical answers are required: one must, *in fine*, resort to simulations. In such situations, significant "practical" progress towards workable solutions is often dependent on the quality of the "information system" which is available and which always must encompass much more than a set of statistical tools, even when packaged into an organic whole. Such considerations justify the second part of the title which is borrowed from the CASE (Computer Aided Systems Engineering) technology discourse: it sees solving a problem (building an information system) as a set of tasks which are grouped into activities which constitute processes. These yield in turn the solution (the information system). In that world tasks require tools, activities, workbenches, and processes, environments [6]. It is claimed that there is a need for analogous concerns and means in the area of computer intensive methods of interest here and a set of "minimal" requirements is given that would provide an adequate environment for the pursuit of such problems. Similar needs arise in certain areas of engineering [1]: the difference

[1] Research support is provided by the Swiss National Science Foundation, Contrat No. 12-36209.92

is mostly with the type of mathematical models that are used.

2 An example: Statistical fit of GBM

GBM is of interest in the financial area because it is intimately linked to the Black-Scholes formula, a formula that allows pricing of an option [4]. To actually use the formula one must obtain an estimate of a parameter which is the diffusion parameter of the GBM which describes the behaviour of the asset supporting the option. A GBM S_t is described [10] implicitly by the stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t,$$

where W is a standard Wiener process, and explicitly by the expression

$$S_t = S_0 e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t}.$$

The statistical problem consists in estimating μ and σ from the observation of a path $f(t)$ of the process S at a finite number of time points

$$t_1 = 0 < t_1 < \cdots < t_n = T.$$

A number of estimators are available [2, 3, 7, 12], but there seems to be little comparative work in settings which are "realistic" (as described in [4] for example), which means in particular that the number of observations is small (between 50 and 200 is "typical"), and often that $T = n$. These constraints raise a number of questions for which there are few analytical answers (it should be stressed that the case of GBM is almost the simplest one could conceive). The recourse is thus simulations. Most methods known so far, and in particular those mentioned here, are, at best, supported by partial simulations which are usually limited to the method presented, and which avoid comparisons with other methods. No systematic statistical investigations exist, which is easily understood, given the complexity of the estimators considered.

One possible method of estimation of μ and σ consists in computing the Radon-Nikodým derivative of the law of S with respect to that of $S_0 + \sigma W$, and of deriving from it estimators which are then "discretized at the observations" [2]. One gets

$$\hat{\mu}_{ML} = \frac{1}{T} \sum_{i=1}^{n} \frac{f(t_i) - f(t_{i-1})}{f(t_{i-1})}$$

$$\hat{\sigma}^2_{QV_2} == \frac{1}{T} \sum_{i=1}^{n} \left[\frac{f(t_i) - f(t_{i-1})}{f(t_{i-1})} \right]^2$$

These estimators are, in general, sums of independent, non identically distributed random varaibles whose law is not exactly expressible analytically. So typically, one must compute moments and derive an asymptotic result. Such calculations require that high order moments (order six in this case) be evaluated. To that end one introduces expressions of the form

$$Ee^{kN_i} = S^{(i)}_{k,\frac{k(k-1)}{2}},$$

where

$$S^{(i)}_{k,l} = e^{k\nu_i + l\sigma_i^2}$$

and N_i is a normal random variable with parameters $\mu_i = (\mu - \frac{\sigma^2}{2})(t_i - t_{i-1})$ and $\sigma_i^2 = \sigma^2(t_i - t_{i-1})(\nu_i = \mu_i + \frac{\sigma_i^2}{2})$. A typical expression is then

$$V(\hat{\sigma}^2_{QV_2}) = \frac{1}{T^2} [(S_{4,6} - S_{4,2}) -$$

$$4\,(S_{3,3} - S_{3,1}) + 4\,(S_{2,1} - S_{2,0})]$$

Here is a list of what a systematic simulation should yield to allow evaluation of such estimators. First, one should distinguish the case of an exact model and that of a model which is approximate. For an exact model, at least the following questions should be answered:

- *Does the value of the parameter to be estimated influence the quality of the estimator?*

  It would indeed not be surprising if very small or very large values of the parameters to be estimated would influence, positively or negatively, the quality of some of the estimators to be considered.

- *What is the influence of the number of observations on the quality of the estimators?*

  What is meant by "number of observations" can be many sided: it may be the absolute number of observations, but it also may be the density of observations (absolute number over time observed, or number per unit of time). In [12] it means four strongly typed observations per day: the question then becomes, how many days?

  The question may also depend on the type of statistical result expected: the number of observations required to obtain a good estimator may be less than that necessary to a validation of the fit. If one's only recourse is a central limit result, when (in terms of absolute numbers or density) does this limit effect take place?

  One may finally ask for "optimal" combinations to insure "overall quality", such as absolute number together with a given duration.

- *Does the regularity of observations matter?*

  Does one need observations taken at regular times, or are observations registered when possible sufficient? In the latter case, is there a "minimum time interval" beyond which estimators become useless?

- *Are there better methods of estimation?*

  In other words can one produce prescriptions for estimation which ensure "quality" of the results?

- *What is the law of the price of the option? Is it sensitive to the estimation procedure, or to any of the potentially disrupting factors?*

  It should be clear that one would need in practice some kind of confidence interval for the price!

In case of a process which does not behave according to the model, a number of obvious questions come to mind. Here are a few:

- *Are the estimators robust?*

  One could ask for the kind of robustness which is expected: the really important one would seem to be that of the law of the price! An associated question would be: are the validation procedures sufficient to at least alert the user to a "departure" from the model, such as a process with sample paths which could be produced by geometric Brownian motion, but which, in reality, are not?

- *Are there procedures which could be used to detect, or to adapt the statistical procedures to, a change in the model?*

  The simplest case would be, for geometric Brownian motion, a change in the values of the drift and the diffusion parameters.

3 A wish list of components for an adequate environment

Evaluation of statistical methods in finance should be performed as a two stage procedure: during the first, one would only be concerned with the purely statistical performance of the method, that is one would want to make sure the method is statistically sound. During the second stage, one would want to check that the method works well for the financial analyst (not the statistician). The latter requires that one has access to databases with financial information, and that a prerequired set of statistical operations be performed. A flexible environment would accept commands which list the operations and the data, and carry out the retrievals and the computations. This is a purely a technical matter for a computer expert. Only the first stage is of interest here.

In the chosen example, there are many ways to estimate the parameters and a number of "dimensions" according to which the evaluation of these estimates should be carried out. The dimensions correspond to the questions raised in section 2. Practically one carries out the simulation as follows.

One begins with simulations of the process (GBM here). To that end one must have at least two tools: an "augmented" random "objects" generator and tools to manage the results of the simulations. Traditional random "objects" generators simulate "objects" whose complexity is that of a random variable (random numbers generators). For finance one must be able to simulate well at least paths of diffusions with state spaces strictly smaller than the real line (assets typically do not have negative values). As shown with the expert systems ADAGIO and PRESTO [9]*(PRESTO is an expert-system which performs automatic generation of complete Fortran programs solving Stochastic Differential Systems, from data provided by a user supposed to have no prerequisite knowledge either in Numerical Analysis of these systems, nor in programmation)*, a useful generator must be coupled with an "AI language" (Lisp in PRESTO) and a symbolic manipulator (REDUCE in PRESTO). In fact, it would be extremely useful to have, among the capacities provided by the symbolic manipulator, facilities which automate stochastic calculus, in the spirit of [8]. Furthermore, the simulator should come with "automatic" tools to check the quality of the paths (if an estimate is computed on a path, one must make sure that what is observed is the behavior of the estimator, and not the behaviour of the simulated path). "Exhaustive" simulation of paths of stochastic processes requires on the other hand that one benefits from facilities to manage the versions, such as "semi-automatic" labeling of files, recording of seeds, and so forth. One should then be able to browse "easily" through these simulations.

Once the paths are available, one needs a "sampler" for at least two purposes. It has been argued that time is an important element for the statistics of financial models. One should thus be able to test the potential estimates against the possible time dimensions as described above. But also some estimation procedures may require specific time sampling. For example, the estimation procedure investigated in [12] requires the first and last daily values of the asset, as well as the largest and the smallest during the day. Thus, to extract from a simulated path different types of samples and associated caracteristics should be an easy operation. Finally, since financial data is "historical" data, the basic assessment technique will eventually be the bootstrap [5] or an adaptation of it (it is thus necessary to produce, from a sampled path, the law of $\hat{\sigma}$ so that the law of the price may be exhibited).

In the area of diffusions many complex objects, such as stochastic integrals, require numerical approximations and it would be useful to have those pre-programmed with quality algorithms as it seems clear that numerical quality is essential for the successful implementation of these rather complicated procedures. Furthermore certain estimation techniques such as filtering [11] ultimately require that numerical schemes for ordinary differential equations be used.

Of course a large array of "ordinary" statistical techniques should be available (for density estimation, for example). These, as hinted in section 2, also require a symbolic calculator to calculate moments explicitly (see the formula for the variance), and other similar calculations. The user of the system should have facilities to enrich and complete it with his or her favourite techniques (access to programming languages and expert systems shells). Reports should be easy to produce (integration of facilities for "intelligent" graphic presentations).

At the present time, tools are available. One needs workbenches and environments!

References

[1] C.R. Baker and A.F. Gualtierotti, Likelihood-ratio detection of stochastic signals, in Advances in Signal Processing, V. Poor and J.B. Thomas, Eds., JAI Press, Greenwhich, CT (1993) 1-34

[2] L. Cedro and A.F. Gualtierotti, Ajustement statistique en finance: le cas du mouvement Brownien géométrique, $XXVI^e$ Journées de Statistique, ASU, Neuchâtel (1994), 188-191

[3] M. Chesney, R.J. Elliott, D. Madan and H. Yang, Diffusion Coefficient Estimation and Asset Pricing when Risk Premia and Sensitivities are Time Varying, Mathematical Finance 3 (1993), 85-99

[4] J.C. Cox and M. Rubinstein, Option Markets, Prentice-Hall, Englewood Cliffs, NJ (1985)

[5] B. Efron and R.J. Tibshirani, An Introduction to the Bootstrap, Chapman and Hall, New York (1993)

[6] A. Fuggetta, A classification of CASE technology, Computer 26, No. 12 (1993), 25-38

[7] L.P. Hansen and J.A. Scheinkman, Back to the Future: Generating Moment Implications for Continuous Time Markov Processes, Preliminary Report, The University of Chicago (1992)

[8] P.E. Kloeden and W.D. Scott, Construction of stochastic numerical schemes through Maple, MapleTech 10 (1993) 60-65

[9] J. Leblond and D. Talay, Simulation of diffusion processes with PRESTO. Building systems like PRESTO with ADAGIO, Cahiers du CERO 32 (1990), 121-134

[10] R. S. Liptser et A. N. Shiryayev, Statistics of Random Processes, Springer, New York, NY (1977)

[11] P.S. Maybeck, Stochastic Models, Estimation and Control, Academic Press, New York (1982)

[12] L.C.G. Rogers and S.E. Satchell, Estimating Variance from High, Low and Closing Prices, The Annals of Applied Probability 1 (1991), 504-512

[13] P.J. Rousseeuw and A.M. Leroy, Robust Regression and Outlier Detection, Wiley, New York (1987)

On Calculating The Distribution of Independent Trials With Changing Probability of Success

Charles R. Katholi, Ph.D.
Karan P. Singh, Ph.D.
and
Alfred A. Bartolucci, Ph.D.

Department of Biostatistics
School of Public Health
University of Alabama at Birmingham
Birmingham, AL 35294-2030

Abstract: The distribution of independent Bernoulli trials is investigated in the case where the probability of success is different at each trial. Expressions for the factorial moments and cumulants are given. These expressions are used to construct the closed form of the probability mass function. The method is shown to be ill-conditioned and Tikhonov regularization is used to compute the probabilities. Formulas for the cumulants and moments are also developed and the probability function approximated by an expansion in the orthogonal polynomials associated with a Binomial distribution.

1. Introduction: In this report, three methods for the computation of the probability mass function of the random variable X, which counts the number of successes in N independent trials, will be considered. In section 2, a direct approach based on exhaustive enumeration will be considered. It will be shown to be impractical in all but the simplest cases. In section 3, formulas for the factorial moments are given. These are used with the formula of Laurent [4] to give a closed form representation of the probability mass function. It is shown that this approach is very ill-conditioned, but that good results can be obtained by Tikhonov regularization. In section 4, an alternative approach based on using moments to approximate the probability mass function by an expansion in orthogonal polynomials is presented. It is found that this approach becomes ill-conditioned as higher moments are used, but that it gives good results in general. Finally, in section 5, a table of results is given for a number of tests of the methods and some comments are made. In what follows, the binomial coefficients will be denoted by C(n,k), vectors by small letters underlined and matrices by capital letters. Small letters with subscripts will denote the elements of vectors and matrices where appropriate.

2. The Direct Method: In this section a formal solution to the problem is presented and analyzed as an approach to computing the probability mass function. Let p_i be the probability of a success on the i-th trial and let k be the number of successes in N trials. Let $I_{C(N,k)}$ be the set of $\underline{z} \in \Re^N$ such that (i). $z_j \in (0,1)$, for j=1,2,...,N and (ii),

$$\sum_{j=1}^{N} z_j = k$$

Then the probability , Pr(X=k), that the random variable X equals k is given by,

$$\sum_{z^{(i)} \in I_{C(N,k)}} \left(\prod_{j=1}^{N} [z_j^{(i)} p_j + (1-z_j^{(i)})(1-p_j)] \right)$$

Each value of the probability mass function requires the summation of C(N,k) products each of which can be expressed as N-1 multiplications. In addition, the computation requires C(N,k)-1 additions. Thus the total number of floating point calculations is (N-1)C(N,k)+C(N,k)-1 for any value of k. Summing over k yields an operations count of $N2^N$ - (N-1) = $O(N2^N)$, so that although simple fast algorithms exist to generate the set of all combinations, the exponential complexity class of the algorithm makes this unfeasible except at the tails of the distribution and for small N. In the evaluation of the methods developed in sections 3 and 4 we will use this calculation procedure to estimate the probabilities for comparison purposes.

3. The Probability Mass Function in Terms of The Factorial Moments: Noting that the factorial moments of a discrete random variable X, X ∈ {0,1,2, ...,N}, with probability mass function, f(x), are defined by the equation,

$$\mu_{[r]} = \sum_{x=0}^{N} x(x-1)\ldots(x-(r-1))f(x)$$

it was shown by Laurent [4] that f(x) has the equation,

$$f(x) = \sum_{j=x}^{N} (-1)^{x+j} C(j,x) \frac{\mu_{[j]}}{j!}$$

The derivation of this result is simple and instructive. If the defining equation for the r-th factorial moment is divided by r!, and r is varied from 0 to N, the resulting system of N+1 linear equations for f(x) is upper triangular with ij-th element equaling C(j-1,i-1) when $j \geq i$ and 0 when j<i.(note that i,j=1,2,...,N+1). It is easily seen that the columns of this matrix are just the rows of Pascal's triangle. The elements of the inverse matrix are just $(-1)^{i+j}$ times the elements of this matrix and so Laurent's formula follows immediately. Examination of this formula reveals potential problems in the computations. In particular, the coefficients of the quantities $\mu_{[r]}/r!$ grow rapidly with N and alternate in sign. In order for the resulting sum to be small cancellations must occur and so it is unlikely that the function can be calculated with good relative precision. The fact that the coefficient matrix has positive elements and is upper triangular suggests solving for the values of f(x) by back substitution. Unfortunately, the matrix is very ill-conditioned with condition number $K_1 = 2^{2N}$. Thus assuming that the quantities $\mu_{[r]}/r!$ can be found, they will be subject to rounding error and we will be considering a classic discrete ill-posed problem. We shall see that this problem can be successfully solved by application of Tikhonov regularization. If we define the factorial cumulants in a manner analogous to the usual cumulants and denote the r-th such quantity by $K_{[r]}$, it can be shown that for the distribution of interest,

$$K_{[r]} = (-1)^{r+1}(r-1)! \sum_{j=1}^{N} p_j^{\,r}$$

Next let $w_r = \mu_{[r]}/r!$ and $v_r = K_{[r]}/r!$ so the w_r can be generated from the v_r by the following recursion,

$$w_{r+1} = \frac{1}{(r+1)} \left\{ w_r v_1 + \sum_{j=1}^{r} (j+1)\, v_{j+1} w_{r-j} \right\}$$

for $r \geq 0$. Combining these two equations yields,

$$w_{r+1} = \frac{1}{(r+1)} \sum_{j=0}^{r} (-1)^{j} \left[\sum_{m=1}^{N} p_m^{\,j+1} \right] w_{r-j}$$

As indicated above, the system of equations for the probability mass function, given the factorial moments becomes increasingly ill-conditioned as N increases. For this reason we apply the method of Tikhonov regularization and restate the problem as a constrained least squares problem:

$$\min_{\underline{f}_\lambda \in \mathbf{R}^N} \|A\underline{f}_\lambda - \underline{\mu}\|_2^2 + \lambda^2 \|\underline{f}_\lambda\|_2^2$$

subject to the constraints,

$$\forall j (0 \leq j \leq N), f_j \geq 0 \; ; \; \sum_{j=0}^{N} f_j = 1$$

The matrix A in these equations is the original upper triangular matrix for the system with the first row and column deleted. The idea of Tikhonov regularization is to choose a suitable value of λ by some criterion. A number of ways of choosing this parameter are described in Hansen [2]. We have considered one of those and also one of our own which is particular to this problem. For reference purposes, these will be denoted by

(1). The Generalized Cross Validation Method (GCV) of Golub, Heath and Wahba [1].

(2). MSRE in which the Mean Square Relative Error is calculated by comparing the factorial moment solution to a few "true" values calculated by the direct method at each end of the solution vector.

It should be noted in this context that even when N is fairly large, the first few values of the probability mass function in each tail of the distribution are easily calculated. In either case, a 1-dimensional nonlinear optimization problem for λ is solved which requires repeated solution of the following constrained linear least squares problem:

Let $\underline{\mu}$ be the N-vector with components $\mu_{[1]},\ldots,\mu_{[N]}/N!$ and I_N be the N x N identity matrix, then for each λ we solve,

$$\begin{vmatrix} A \\ \lambda I_N \end{vmatrix} \underline{f}_\lambda = \begin{vmatrix} \underline{\mu} \\ \underline{0} \end{vmatrix}$$

subject to the constraints

$$f_{\lambda_j} \geq 0 \;,\; j=1,\ldots,N$$

$$\sum_{j=1}^{N} f_{\lambda_j} = 1-\prod_{j=1}^{N}(1-p_j)$$

where p_j is the known probability of a success on the j-th trial.

The results of computational experiments with this approach are given in section 5. It will be shown there that both of the methods indicated above for choosing the parameter λ give satisfactory results. However, the GCV method, which requires the computation of the Singular Value Decomposition of the matrix A, appears to give slightly poorer results.

4. Approximation By Expansion In Orthogonal Polynomials: The technique to be used here is applicable to any discrete distribution and will be described in very general terms.

Let f(x) be the discrete distribution to be approximated and let f(x) be defined on the set Ω, $\Omega=\{x_0,x_1,\ldots,x_m\}$. Let p(x) be a second known discrete distribution with domain Ω. Finally let $\{h_j(x),j=0,1,\ldots\}$ be a set of polynomials orthogonal to each other with respect to p(x)'on Ω; that is such that

$$\sum_{x=x_0}^{x_m} p(x)\,h_i(x)\,h_j(x) = \begin{cases} 0 \;, & i\neq j \\ \|h_j\|^2_{p(x)} \;, & i=j \end{cases}$$

By matching moments we shall find coefficients a_0, a_1, a_2, ... such that

$$f(x) \approx p(x)\;[a_0+a_1h_1(x)+a_2h_2(x)+\ldots]$$

If an approximation utilizing the first r terms of this expansion is to be generated, then the following result can be easily derived,

<u>THEOREM:</u> Let $X \in R^{(m+1)x(r+1)}$ and $D \in R^{(m+1)x(m+1)}$ be defined as

$$X = \begin{vmatrix} 1 & x_0 & x_0^2 & \ldots & x_0^r \\ 1 & x_1 & x_1^2 & \ldots & x_1^r \\ 1 & x_2 & x_2^2 & \ldots & x_2^r \\ & & & \ldots & \\ 1 & x_m & x_m^2 & \ldots & x_m^r \end{vmatrix}$$

and $D=\mathrm{diag}(p(x_0),p(x_1),\ldots,p(x_m))$. Let $A = D^{1/2}X$ have QR factorization,

$$A = Q\begin{vmatrix} R_{11} \\ 0 \end{vmatrix} \;,\; R_{11} \in \mathbf{R}^{(r+1)x(r+1)}$$

Then the columns of the matrix $B=D^{-1/2}Q$ are orthogonal with respect to the weight matrix D and are the orthogonal polynomials $h_0(x)$, $h_1(x)$, ... $h_r(x)$ evaluated on Ω. Furthermore, if $\underline{\mu}$ is the vector composed of the 0-th moment and the first r moments of f(x) then the coefficients a_i in the expansion are the solutions to the system of equations $R_{11}^T\,\underline{a} = \underline{\mu}$.

Again formulas for the moments and cumulants of the distribution under study are easily calculated as functions of the known probabilities of success on individual trials. To this end, for each p_j, let $d_{n+1}(j)$ be defined by, $d_1 = p_j$

$$d_{n+1}(j) = p_j[1-\sum_{i=1}^{n} C(n,k)\,d_{n+1-j}(j)]$$

then the cumulants K_r are given by

$$K_r = \sum_{j=1}^{N} d_r(j)$$

The moments are then found from the cumulants by the well known formula,

$$\mu_{r+1} = \sum_{j=0}^{r} C(r,j)\,K_{j+1}\mu_{r-j}$$

The matrix R_{11} tends to become ill-conditioned as r increases because the matrix $D^{1/2}X$ becomes ill-conditioned. The degree of ill conditioning is a function of the

Table I.
Comparison of the directly computed CDF to those obtained by the GCV, MSRE and series approximation method for the case of N=20 and 100 simulations. The table entries are relative differences.

GCV (N=20) 100 trials						
	lower tail			upper tail		
	1%	5%	10%	1%	5%	10%
Max	5.5 x 10^{-5}	2.2 x 10^{-5}	1.3 x 10^{-5}	3.5 x 10^{-7}	3.9 x 10^{-7}	1.3 x 10^{-6}
Q3	3.1 x 10^{-6}	2.9 x 10^{-6}	1.4 x 10^{-6}	8.5 x 10^{-8}	-4.5 x 10^{-9}	1.7 x 10^{-7}
Med	5.4 x 10^{-7}	1.0 x 10^{-7}	3.0 x 10^{-7}	2.3 x 10^{-8}	-5.1 x 10^{-8}	3.5 x 10^{-8}
Q1	-4.9 x 10^{-7}	-5.3 x 10^{-7}	-1.4 x 10^{-7}	-3.1 x 10^{-9}	-1.6 x 10^{-7}	-3.6 x 10^{-8}
Min	-6.0 x 10^{-5}	-8.9 x 10^{-6}	-8.5 x 10^{-6}	-6.2 x 10^{-8}	-7.3 x 10^{-7}	-5.9 x 10^{-7}

MSRE (N=20) 100 trials 4 end points						
	lower tail			upper tail		
	1%	5%	10%	1%	5%	10%
Max	1.1 x 10^{-7}	2.1 x 10^{-9}	3.4 x 10^{-8}	1.4 x 10^{-9}	8.9 x 10^{-10}	4.4 x 10^{-9}
Q3	1.4 x 10^{-8}	5.9 x 10^{-9}	4.9 x 10^{-9}	1.6 x 10^{-10}	5.3 x 10^{-11}	4.4 x 10^{-10}
Med	1.1 x 10^{-9}	1.5 x 10^{-10}	4.9 x 10^{-10}	2.9 x 10^{-11}	-1.6 x 10^{-10}	5.4 x 10^{-11}
Q1	-6.6 x 10^{-9}	-3.6 x 10^{-9}	-1.5 x 10^{-9}	-4.7 x 10^{-11}	-3.6 x 10^{-10}	-2.7 x 10^{-10}
Min	-5.7 x 10^{-7}	-4.7 x 10^{-8}	-7.6 x 10^{-8}	-6.3 x 10^{-10}	-2.9 x 10^{-9}	-2.9 x 10^{-9}

SERIES (N=20) 100 trials						
	lower tail			upper tail		
	1%	5%	10%	1%	5%	10%
Max	4.1 x 10^{-2}	5.4 x 10^{-4}	1.9 x 10^{-3}	6.3 x 10^{-4}	3.4 x 10^{-4}	3.8 x 10^{-4}
Q3	2.1 x 10^{-3}	-2.0 x 10^{-4}	1.0 x 10^{-4}	-1.5 x 10^{-4}	7.1 x 10^{-5}	6.3 x 10^{-5}
Med	3.7 x 10^{-4}	-7.9 x 10^{-4}	-1.5 x 10^{-4}	1.6 x 10^{-5}	3.0 x 10^{-5}	2.0 x 10^{-5}
Q1	-1.3 x 10^{-3}	-2.1 x 10^{-3}	-5.9 x 10^{-4}	-1.6 x 10^{-6}	1.3 x 10^{-5}	-6.5 x 10^{-7}
Min	-2.6 x 10^{-2}	-2.8 x 10^{-2}	-6.3 x 10^{-3}	-1.4 x 10^{-4}	-1.3 x 10^{-4}	-4.4 x 10^{-4}

distribution p(x). The degree of ill-conditioning can be controlled by the choice of r. It is expected that the quality of the approximation will improve as the number of moments r increases. In fact, if m is finite, then using m moments will give an exact result. On the other hand, as the number of moments increases, so does the condition number of R_{11} and so a reasonable balance between accuracy and conditioning must be found. Since R_{11} is upper triangular, the condition number is easily calculated to help in this decision.

5. Computational Results and Conclusions: All computations presented were performed in IEEE Binary Rounded Double Precision floating point arithmetic on an Intel Pentium processor. The codes were written in WATCOM FORTRAN 77[32] and run under the OS/2 2.11 operating system. The constrained least squares problems were solved using the codes of Hanson and Haskell, TOMS Algorithm 587 [3]. In all cases, the values of the cumulative distribution function resulting from the computed probability mass functions found by the methods of sections 3 and 4 were compared to like values found by the direct method of section 2. The values presented in Table I are for the relative differences between the computed values. It should be noted that the values computed directly are also subject to error. In particular, although a value calculated directly is an unbiased (with respect to the distribution of the rounding errors) estimate of the true value, its variance grows as N grows and so any confidence interval grows as well. Thus we will refer to these as relative differences in the computed values but not as relative errors. Rather than give mean relative differences in Table I, we give a five number display which includes the extremes, the quartiles and the median. In addition, we give results for "nominal" 1%, 5% and 10% critical points in both tails of the distribution. Since the distribution is discrete, these levels are not exact and represent the relative difference at the point on the CDF which is closest to the indicated probability level.

Results are presented for the case of N=20 for the GCV method and for the MRSE method. These are based on 100 randomly generated sets of probabilities of success. For the MSRE method, results are given for the cases of 4 directly calculated values used at each end to estimate the Mean Square Relative Error. The MSRE method for N=25 and 4 points at each end gave similar results and is not shown due to space limitations. The values in the table indicate that all methods of choosing the lambda yield satisfactory results while the MSRE method gives results which are slightly better than those found by the GCV method. The advantage of the GCV method is that it requires no direct calculations of the tails of the distribution. The disadvantage is that it requires the calculation of the Singular Value Decomposition (SVD) of one matrix. In our experience, the extreme ill-conditioning of the matrix caused the SVD code to fail when N approached about 50. It should be noted that this is the point at which the elements, C(N,k), of the matrix can no longer be represented exactly in the floating point system.

For the orthogonal expansion approximation method, results are given for N=20 using 8 moments and the Binomial distribution with p chosen so the its mean matches that of the target distribution. Again results are given for 100 simulations. Like any such expansion, the values in the tail area are particularly sensitive to the number of moments used. However, the simulation results indicate that even though some of the probability estimates in the tails can be negative (and small) the values of the CDF at the approximate 1%, 5% and 10% levels are not badly effected. The overall results can be improved slightly if the most extreme few probabilities are calculated directly.

In conclusion we note that any of the methods described can yield values of the CDF which are satisfactory for practical work. The method based on the factorial moments is more computationally intensive and can be expected to yield more accurate results. The approximation method yields less accurate results in general unless all moments are used in which case the results are comparable. The approximation method was tested for randomly chosen p_j on (0,1). Intuitively, we would expect it to perform better in situations where the p_j are different but are on a narrower interval.

References:

[1]. Golub, G.H., Heath,M.T. and Wahba, G.,*Generalized Cross-Validation As A Method For Choosing A Good Ridge Parameter*, Technometrics, 21 (1979), pp215-223.

[2]. Hansen, P.C.,*Analysis Of Discrete Ill-Posed Problems By Means Of The L-Curve*, SIAM Review, 34,#4 (1992),pp561-580.

[3]. Hanson, R.J. and Haskell, K.H.,*ALGORITHM 587,Two Algorithms For The Linearly Constrained Least Squares Problem*, ACM Transactions on Mathematical Software, 8,#3 (1982), pp323-333.

[4]. Laurent, G.A.,*Probability Distributions, Factorial Moments, Empty Cell Test*, Classical and Contagious Discrete Distributions, Proc. Int. Symp. Classical and Contagious Discrete Distributions, Montreal, Pergamon Press, New York (1965), pp 437-442.

Bayesian Estimation using the Gibbs Sampler for the Inhibition/Promotion Cancer Chemoprevention Experiment

C. Hsu and J. Michael Hardin

Civitan International Research Center
Department of Biostatistics
University of Alabama at Birmingham

Abstract:

Kokoska (1987) suggested a set of maximum likelihood estimators relevant to the analysis of the Inhibition/Promotion (I/P) mammary cancer chemoprevention experiment. This set of estimators has been extended and studied in various detail in a number of related papers, Kokoska (1988a, 1988b), Hsu (1990), Kokoska, Hardin, Hsu, and Grubbs (1993). Often, however, investigators have some prior knowledge of a compound tested in such experiments due to its chemical structure and similarity to related compounds. In such situations, experimenters often wish to exploit this prior knowledge in order to reduce the costs of experimentation. Thus, this paper examines Bayesian estimators for this purpose and numerical algorithms, based on the Gibbs sampler (Gelfand and Smith, 1990) and the rejection method (Smith and Gelfand, 1992), with which to compute the posterior distribution. The methodologies are illustrated with experimental data taken from Grubbs (1993).

1. Introduction.

The Inhibition/Promotion (I/P) Cancer Chemoprevention Experiment is designed to investigate the effect of compounds that can be given in the diet on incidence rates of cancer. These experiments are often administrated by the National Cancer Institute. The primary purpose of the experiment is to isolate and identify potential cancer inhibiting or promoting substances in human. Variables of interest in these experiments are the incidence of tumors in the animals, the number of tumors per animal, and the rate at which tumors develop. The Chemoprevention Branch, Division of Cancer Prevention and Control, in the National Cancer Institute has issued guidelines for statistical analysis such as log-rank test and Armitage test. However, difficulty in analyzing these experiments may occur due to the fact that the experiment is terminated before all the induced tumors have been observed (i.e., right censored data). Therefore, a confounding of fewer observed tumors in treatment group compared to control could be the result of a decreased number of induced tumors, a decreased growth rate of tumor, or both occur. The problem results from the fact that the number of induced tumor (M) in each animal is dependent upon the time to tumor detection (T). Current statistical methods do not account for this confounding since they do not test the number of induced tumor and the time to tumor detection simultaneously.

Kokoska (1987) suggested a set of maximum likelihood estimators relevant to the analysis of the mammary cancer chemoprevention experiment. This set of estimators have been extended and studied in various detail in a number of related papers, Kokoska (1988a, 1988b), Hsu (1990), Kokoska, Hardin, Hsu, and Grubbs (1993). In this paper the basic idea of Kokoska's method will be reviewed.

2. Mathematical Model of Kokoska's Approach

Kokoska proposed modelling the number of induced tumors, M, as a Poisson distribution, and the time to tumor detection, T, as a gamma distribution. Suppose that a treatment group consists of n animals, and m_i $(i=1, 2, ..., n)$ is the number of promoted tumors in animal i. Let t_{ij} be the observed time to detection of tumor i in animal i $(j = 1, 2, ..., m_i)$, and let $J(t_i)$ be the number of observed tumors for the animal i at the time t_i. Further, denote the mean and variance of X μ_M and σ^2_M, respectively. Let $F(t)$ be the cumulative density function (cdf) of T. Kokoska (1987) has shown that $J(t_i)$ has mean $\mu_M F(ti)$ and variance $(\sigma^2_M-\mu_M)F^2(t_i)$ $+\mu_M F(t_i)$. These result demonstrate mathematically the dependence of the number of detectable tumors at time t_i on the mean number of induced tumors and the time to tumor detection.

The log-likelihood function of J(t) can be shown as below.

$LL(\lambda, \alpha, \beta) = -\lambda \Sigma_{i=1}^{n} F(t_i^*; \alpha, \beta) + s_1\{ln(\lambda) - \alpha ln(\beta) - ln(\Pi(\alpha))\} + s_2(\alpha-1) - s_3/\beta - ln(K)$

where $F(t_i*; \bullet, \bullet)$ denotes cumulative density function of T, and $s_1 = \Sigma_{i=1}^{n} m_i$, $s_2 = \Sigma_{i=1}^{n} \Sigma_{j=1}^{mi} t_{ij}$, and $K = \Pi_{i=1}^{n} m_i!$, and all the animals are sacrificed at the end of the experiment $t*$.

This log-likelihood can be numerically optimized to obtain the MLE.'s of interest using the IMSL FORTRAN library subroutine DBCONF. The mean number of induced tumors per animal can be estimated via the MLE $\hat{\lambda}$. However, the parameter μ, the mean time to tumor detection, is of more biological significance than the estimates of the parameters associated with each of the continuous distributions. An MLE of μ can be easily obtained using the invariance property of MLE's, i.e., $\hat{\mu} = \hat{\alpha}\hat{\beta}$ (Roussas, 1973).

This parametric model has been extended using various assumptions to eight models (Kokoska, 1988; Hsu, 1990; Hardin and Hsu, 1991; Kokoska et al., 1993) for different kinds of data assumptions. In this paper, however, Poisson and gamma distributions for the number of induced tumors in each animal and their times to tumor detection, respectively, are examined in comparison to the estimates using the Bayesian approaches.

3. Bayesian Methods

Since investigators may have some prior knowledge of a compound tested in such experiments due to its chemical structure and similarity to related compounds, they may wish to exploit this prior knowledge to reduce the duration of the experiment or to lessen the number of experimental animals due to the cost of experimentation. This section examines Bayesian estimators, based on the Gibbs sampler (Gelfand and Smith, 1990) and the rejection method (Smith and Gelfand, 1992), are both presented. These techniques are applied to an actual experimental data.

Gibbs sampling has allowed the computation of complicated statistical models based on Bayesian posterior inference. Additionally, the rejection method of Smith and Gelfand is a straightforward sampling-resampling perspective that allows the computation of Bayesian estimators using easily implemented calculation strategies. The methodologies are introduced as follows.

(1) Gibbs sampling method

Suppose that X, Y, and Z are the random variables, and their conditional distributions, $f_{X|Y,Z}(x|y,z), f_{Y|X,Z}(y|x,z), f_{Z|X,Y}(z|x,y)$ are known. If initial values of x_0', y_0' are specified, then a "Gibbs sequence of value" of the random variables, $X_0', Y_0', Z_0', X_1', Y_1', Z_1', ..., X_k', Y_k', Z_k'$, can be obtained iteratively by alternately generating values from

$X_i' \sim f_{X|Y',Z'}(x|Y_i'=y_i', Z_i'=z_i')$ and
$Y_i' \sim f_{Y|X',Z'}(y|X_i'=x_i', Z_i'=z_i')$ and
$Z_i' \sim f_{Z|X',Y'}(z|X_i'=x_i', Y_i'=y_i')$

It turns out that under reasonably general conditions, the distribution of X_K' converges to $f_X(x)$, which is the true marginal of X as $k \to \infty$ (Casella and George, 1992). Thus for k large enough the final observation $X_k'=x_k'$ is effectively a sample from $f_X(x)$. So are the observations $f_Y(y)$ and $f_Z(z)$.

In this paper the conditional probabilities of the parameters of interest are assumed as follows.

$f(\lambda|\alpha, \beta) \sim Gamma(\beta/\alpha, 1)$, and
$f(\alpha|\beta, \lambda) \sim Normal(\beta/\lambda, \lambda)$, and
$f(\beta|\alpha, \lambda) \sim Normal(\alpha\lambda, \lambda)$.

One thousand iterations were made to get the marginal distributions for the parameters, and 10,000 sample sizes were generated.

(2) Rejection Method

For fixed $\underline{x}$, let $f_X(\underline{\theta}; \underline{x}) = l(\underline{\theta}; \underline{x})p(\underline{\theta})$ where $l(\underline{\theta}; \underline{x})$ is the likelihood function of θ and $p(\underline{\theta})$ is the prior distribution of $\underline{\theta}$. If $\underline{\theta}$ is the M.L.E. of $\underline{\theta}$, and $M = l(\underline{\theta}, \underline{x})$. The first step of this method involves the generation of θ from $p(\theta)$ and also the generation u from continuous uniform distribution (0, 1). Second, evaluate the following procedure

$u \le f_X(\underline{x})/(Mp(\underline{\theta})) \Rightarrow accept\ \underline{\theta}$
$u > f_X(\underline{x})/(Mp(\underline{\theta})) \Rightarrow reject\ \underline{\theta}$

Thus a sample of the posterior distribution of the parameter $\underline{\theta}$ can be obtained if the above procedures are applied repeatedly. In this paper uniform distributions were used for the prior distributions of the parameters, α, β, and λ, and 10,000 simulations were generated.

4. Application

In a study (Grubbs, 1993), sixty female Sprague-Dawley rats were randomly divided into 2 groups. In Group 1 the rats were treated by retinoid vehicle, and the rats in Group 2 were treated by RTBE (934 mg/kg of diet). Then the MNU was administrated to every animal. In both experiments, the animals were palpated for the detection of mammary tumors. The investigation was terminated 182 days after the injection of carcinogen. Tables 1 and 2 give the survival times, the numbers of induced tumors, and the times of development of mammary cancer for each group.

5. Discussion

Tables 1 and 2 present the maximum likelihood estimates for the mean number of induced tumors per animal and for the mean time to tumor detection and the corresponding 95% confidence intervals and 95% credibility intervals using classical approach and the Bayesian techniques for each group. Clearly, the estimates $\hat{\mu}_M$ and $\hat{\mu}_T$ using Kokoska approach and the rejection method of Smith and Gelfand are very close; and their 95% confidence intervals and credibility intervals are similar, as well. However, the estimates using the rejection method seem to be a slightly better than that of the classical approach since the credibility intervals are narrower than the corresponding confidence intervals. The estimates of the parameters of interest using the Gibbs sampling techniques are not good compared to the estimates using either the classical or the rejection methods. This might be due to the selection of inappropriate prior conditional distributions for the parameters of interest. Work is currently undergoing to incorporate researchers' experience to obtain better prior conditional densities for the parameters.

Table 1. Estimates and 95% Confidence/Credibility Intervals of the Parameters for Group 1 (Control Group)

		Kokoska's method	Gibbs's Sampler	Rejection Method
λ	Estimate	**15.66**	**8.69**	**15.81**
	95% Confidence /Credibility Interval	**(13.15, 18.46)**	**(5.14, 18.57)**	**(13.30, 17.65)**
μ	Estimate	**185.83**	**241.25**	**187.76**
	95% Confidence /Credibility Interval	**(171.92, 199.75)**	**(26.14, 579.43)**	**(173.05, 203.79)**

Table 2. Estimates and 95% Confidence/Credibility Intervals of the Parameters for Group 2 (Treatment Group)

		Kokoska's method	Gibbs's Sampler	Rejection Method
λ	Estimate	**9.19**	**8.71**	**9.32**
	95% Confidence /Credibility Interval	**(7.46, 11.07)**	**(5.15, 18.50)**	**(7.70, 10.67)**
μ	Estimate	**155.30**	**242.17**	**154.55**
	95% Confidence /Credibility Interval	**(143.03, 167.56)**	**(26.25, 590.10)**	**(138.13, 170.49)**

Acknowledgements

We would like to thank Dr. Clinton Grubbs for allowing us to use the data in this paper.

References

Berger J. O. 1985. Statistical Decision Theory and Bayesian Analysis. 2nd ed. Springer-Verlag. New York.

Casella G. and George E. I. 1992. Explaining the Gibbs Sampler. *The American Statistician.* 46 : 167 - 174.

Gelfand A. E., Smith A. F. M. and Lee T. 1992. Bayesian Analysis of Constrained Parameter and Truncated Data problems Using Gibbs Sampling. *JASA.* 87 : 523 - 532.

Greenhouse J. B. 1992. On Some Applications of Bayesian Methods in Cancer Clinical Trials. *Statistics in Medicine.* 11: 37 - 53.

Grubbs C.J. 1993. Personal Communication.

Hardin, J.M., Hsu, C., Kokoska, S.M., and Grubbs, C.J. 1992: A Generated Program for Cancer Chemoprevention Experiments, Proceedings of Statistical Computing Section, American Statistical Association, American Stat. Asso. 1991, 270-275.

Hsu C. 1990. A Study of Some Techniques for Cancer Chemoprevention Experiments. Unpublished Master Thesis.

Kokoska S. M. 1987. The Analysis of Cancer Chemoprevention Experiments. *Biometrics.* 43 : 525-S34.

Kokoska S. M. 1988a. The Analysis of Cancer Chemoprevention Experiments in Which There Is Interval Censoring. *Applied Mathematics Letters.* 1 : 1-4.

Kokoska S. M. 1988b. Including Data from Early Deaths in the Analysis of Cancer Chemoprevention Experiments. *Applied Mathematics Letters.* 1 : 197-201.

Kokoska S. M. 1988c. The Analysis of Cancer Chemoprevention Experiments in Which There Is Heterogeneous Poisson Sampling. *Applied Mathematics Letters.* 1.

Kokoska, S.M., Hardin, J.M., Grubbs, C.J., and Hsu, C. 1993: The Statistical Analysis of Cancer Inhibition/Promotion Experiments, *Anticancer Research,* 1357-1364.

Press S. J. 1989. Bayesian Statistics. Wiley Interscience. New York.

Ritter C. and Tanner M. A. 1992. Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler.*JASA.* 87 : 861 - 868.

Smith A. F. M. and Gelfand A. E. 1992. Bayesian Statistics without Tears: A Sampling-Resampling Perspective. *The American Statistician.* 46 : 84 - 88.

Tanner M. A. 1993. Tools for Statistical Inference. Springer-Verlag. New York.

Roussas G.G. 1973. A First Course in Mathematical Statistics. Addison-Wesley Publishing Co., Massachusetts.

A Generalized Measure of Dependence with an Application to Molecular Similarity Analysis

Mark A. Johnson, Cheng Cheng, Gerald M. Maggiora and Michael Lajiness
Upjohn Laboratories, Kalamazoo, MI 49007-4940

ABSTRACT. Molecular similarity analysis involves the analysis of data based on complex data types used to represent the information in molecular structures. With many complex data types, we lose many nice features of vector spaces, but retain the concept of proximity. Let X be a random variable with density f defined on a space Ω. Let g be any other density defined on Ω. Define the relative aggregation $\alpha(f|g)$ of f with respect to g by

$$\alpha(f|g) = \frac{\int f^2 g}{\left(\int fg\right)^2}.$$

Suppose $X = (X_1, X_2)$ with marginal densities f_1 and f_2. Define $g = \frac{1}{2}f + \frac{1}{2}f_1 f_2$. Define the dependence coefficient $\delta(X_1, X_2)$ by $\delta(X_1, X_2) = \alpha(f|g)/\alpha(f)$ where $\alpha(f)$ is the relative aggregation of f with respect to itself. We show that if f is the bivariate normal density, then the δ-coefficient varies monotonically with the correlation coefficient. The δ-coefficient can be estimated using random quadrat sampling when a suitable proximity measure is defined on Ω. Two representations used in computing molecular similarity are shown to have a high delta coefficient.

1. Introduction

Statisticians continue to encounter increasingly complex data types. In our application of molecular similarity analysis to drug discovery research, examples include binary vectors representing the presence or absence of up to 300 molecular fragments, labeled graphs representing the bonding structures of molecules, and scalar fields in $\mathbb{R}^3$ for representing the electrostatic fields of molecules (Johnson, 1989, Johnson and Maggiora, 1990). Problems associated with high dimensionality abound, and in some cases, we even lose the natural definitions of such concepts as coordinates, location, and linear transformations. One important concept that remains is proximity. If two objects are represented by the same data type, we can virtually always measure how similar one is to the other.

Recently, Cheng and Johnson (1994a,1994b) proposed the concept of relative aggregation coefficients as a method of developing statistical inference on probability spaces in which a proximity measure has been defined. Here we illustrate the use of relative aggregation coefficients in developing a general measure of dependence between two random variables. Although our approach generalizes directly to arbitrary probability spaces, the discussion will be limited to Euclidean spaces. After defining relative aggregation coefficients and presenting a moment estimator for them, we develop a coefficient of dependence and show its relationship to the bivariate normal correlation coefficient. We then compute the dependence coefficient for two high-dimensional vector representations used for measuring molecular similarity.

2. Relative Aggregation Coefficients

Let f and g be probability density functions defined on $\mathbb{R}^k$ such that $\int f^2 g$ exits. Then the relative aggregation coefficient (RAC) $\alpha(f|g)$ of f with respect to g is defined by

$$\alpha(f|g) = \frac{\int f^2 g}{\left(\int fg\right)^2}.$$

If $g = f$, then we write $\alpha(f)$ for $\alpha(f|g)$, and we all $\alpha(f)$ the self aggregation coefficient of f.

Some insight into aggregation coefficients is gained by viewing these integrals as moments of $f(Z)$ where g is the density of Z. Write $E_g[f^i(Z)]$ for $\int f^i g$ and call it the i'th-relative moment of f with respect to g, or simply the i'th self moment of f if $g = f$. Then the RAC of f with respect to g is simply the second relative moment of f with respect to g divided by the square of the corresponding first relative moment. It follows immediately that $\alpha(f|g) \geq 1$.

What would make this ratio large? Consider any other density h for which $\int fh < \epsilon \int f^2$. Let g be the mixture $pf + qh$ where $p + q = 1$. Then $\int f^2 g > p \int f^3$, and $\int fg < \int f^2(p + q\epsilon)$. It follows that $\alpha(f|g) > p\alpha(f)/(p + q\epsilon)^2$ which goes to $p^{-1}\alpha(f)$ as $\epsilon \rightarrow 0$. Since $\alpha(f) \geq 1$, we can always find a g so as to make $\alpha(f|g)$ arbitrarily large.

3. A Coefficient of Dependence

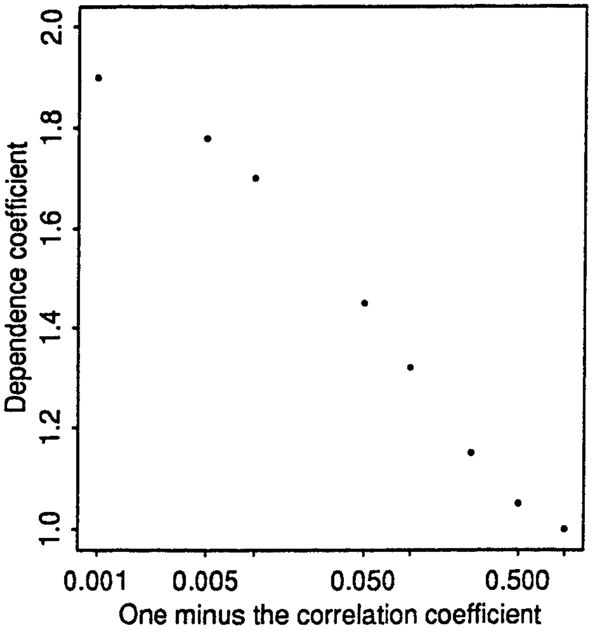
Let X and Y be two random variables defined on $\mathbb{R}^{k_1}$ and $\mathbb{R}^{k_2}$ where $k_1 + k_2 = k$. Let f denote the joint density of (X, Y), and let f_1 and f_2 denote the respective marginals of f. Let h be the product density $f_1 f_2$. Then X and Y are independent if and only if $f = h$. Define $g = \frac{1}{2}f + \frac{1}{2}h$, and define the dependence coefficient $\delta(X, Y)$ by

$$\delta(X,Y) = \frac{\alpha(f|g)}{\alpha(f)}.$$

Clearly if $f = h$, then $\delta(X, Y) = 1$. On the other hand, we see from the preceding section that $\delta(X, Y) \simeq 2$ whenever $\int fh \simeq 0$.

Figure 1 plots the dependence coefficient in the case f is the bivariate normal for various values of the correlation coefficient. A distinct monotonic relationship is obtained. The correlation coefficient in the figure could be replaced by its absolute value as aggregation coefficients are invariant under a particular subclass of linear transformations on $\mathbb{R}^k$, as is now demonstrated.

Figure 1. Dependence coefficient versus the log of one minus the correlation coefficient.



Define T by

$$T = \begin{bmatrix} T_x & 0 \\ 0 & T_y \end{bmatrix}$$

where T_x and T_y are square-nonsingular matrices with k_1 and k_2 rows respectively. Let x_o and y_o be fixed vectors of length k_1, and k_2. Then the density $\widehat{f}$ of $T(X - x_o, Y - y_o)$ is given by

$$\begin{aligned} \widehat{f} &= |(T^{-1})| f(T^{-1}(X - x_o, Y - y_o)) \\ &= |T_x^{-1}||T_y^{-1}| f(T_x^{-1}(X - x_o), T_y^{-1}(Y - y_o)). \end{aligned}$$

It then follows that the marginal densities of $\widehat{f}$ are given by $|T_x^{-1}| f_1(T_x^{-1}(X - x_o)$ and $|T_x^{-1}| f_1(T_x^{-1}(X - x_o)$. Straight forward calculations give $\delta(X, Y) = \delta(T_x(X), T_y(Y))$.

4. A Consistent Estimator

Let X, Y, f, f_1, f_2, and h be as defined. We seek a consistent estimator of $\delta(X, Y)$ which is a ratio of two RACs. Since the denominator is bounded away from zero, it follows that the ratio of consistent estimators of the numerator and denominator of $\delta(X, Y)$ is a consistent estimator of $\delta(X, Y)$. A consistent moment estimator of a RAC is presented in Cheng and Johnson (1994c) elsewhere in this volume. Briefly, it is constructed as follows: Let $S = \{(x_1, y_1), ..., (x_N, y_N)\}$ be a dataset of N independent samples from f, and let $z_1, ...z_m$, be m independent samples from density g where g is any other density defined on $\mathbb{R}^k$. Let d be any proximity measure defined on $\mathbb{R}^k$. Define

$$B_r(z) = \{(x, y) | d((x, y), z) < r\}.$$

and define $n_r(z_i)$, $i = 1, ..., m$, to be the cardinality of the set

$$\{(x, y) | (x, y) \in B_r(z_i)\, ,(x, y) \in S\, , \text{and} (x, y) \neq z_i\}.$$

Let $\overline{x}_r$ and s_r^2 be the sample mean and variance of $n_r(z_i)$, $i = 1, ..., m$, and define $A_r = s_r^2 / \overline{x}_r$. Then Cheng and Johnson show that

$$\widehat{\alpha}(f|g) = \frac{N}{N-1} \times \frac{s_r^2 - \overline{x} + \overline{x}^2}{\overline{x}^2}$$

is a consistent estimator of $\alpha(f|g)$ under the assumption that

$$\int_{B_r(z)} f(t)dt = f(z) \int_{B_r(z)} dt + o(\int_{B_r(z)} dt).$$

(The optimal estimation of $\alpha(f|g)$ is the subject of another study.)

In spatial statistics, the neighborhood $B_r(z_i)$ is called a quadrat centered at z_i, and gives rise to the term "random quadrat sampling", when z_i represents the outcome of a random variable. Random quadrat sampling requires the definition of a proximity measure. Interestingly, proximity measures do not figure into the definition of RACs, but often enter into the picture when RACs are being estimated. With a consistent estimator now in hand, all that remains is to clarify how one defines quadrat sampling with respect to densities f and h.

As in the example that follows, usually one has proximity measures d_1 and d_2 associated with X and Y and must construct a proximity measure d for (X, Y). There are many ways. The following definition is convenient from a computational standpoint

$$d((x_1, y_1), (x_2, y_2)) = \max[d_1(x_1, x_2), d_2(y_1, y_2)]. \tag{1}$$

To illustrate this convenience, write $z = (u, v)$ where u and v and k_1 and k_2-dimensional vectors. Define $B_{r,1}(u) = \{x | d_1(x, u) < r\}$ and $B_{r,2}(u) = \{x | d_2(x, u) < r\}$. Now consider our problem in estimating the numerator of $\delta(X, Y)$. We must count the number of points in S which fall in the quadrat when half of the time the center of the quadrat is drawn according to the joint density f and the other half of the time the center is drawn from the product density h. In either case, the cardinality $n_r(z_i)$, $i = 1, ..., m$, when d is defined by equation 1, is simply the cardinality of the intersection of the following two sets:

$$\{(x, y) | x \in B_{r,1}(u_i), (x, y) \in S, \text{and } x \neq u_i\}.$$

and

$$\{(x, y) | y \in B_{r,2}(v_i), (x, y) \in S, \text{and } y \neq v_i\}.$$

We assure that z_i, $z_i = (u_i, v_i)$, is drawn according to f, by drawing z_i at random from S. We assure that z_i is drawn according to h by drawing u_i at random from the set $\{x | (x, y) \in S\}$ and then drawing v_i at random from the set $\{y | (x, y) \in S\}$.

5. An Example

There is an increasing use of molecular similarity measures in the pharmaceutical industry (Johnson and Maggiora, 1990). Similarity searching is a frequent application in which one searches a large databases of molecular structures for structures similar to some query structure of pharmaceutical interest. The desire is to find related compounds in the database which might also be expected to be of related interest. See Willett (1987) for detailed coverage of the issues and many of the proximity measures being used in this regard. One expects most of these proximity measures to be highly related. In this example, we study the relationship between two proximity measures, topological index (TI) distance and fragment representation (FR) similarity, used at our company for fast similarity searching.

In mathematical chemistry, a topological index is simply a number calculated on the bonding structure of a molecule. A simple count of the number of atoms serves as an example of a topological index, although most topological indices are considerably more sophisticated. The representation for our TI distance is the first 10 principal components of 90+ topological indices (Basak, et al., 1988). The TI distance is simply the Euclidean distance in $\mathbb{R}^{10}$. Our fragment representation of a molecular structure is a binary vector $\mathbf{x}$ in which each bit represents the presence or absence of at least one structural fragment (connected substructure) in a fragment group. Over 300 groups of fragments are used. The similarity measure is the Jacard coefficient (usually called the Tanimoto coefficient in chemistry) defined by $\mathbf{x}'\mathbf{y}/(\mathbf{x}'\mathbf{x} + \mathbf{y}'\mathbf{y} - \mathbf{x}'\mathbf{y})$.

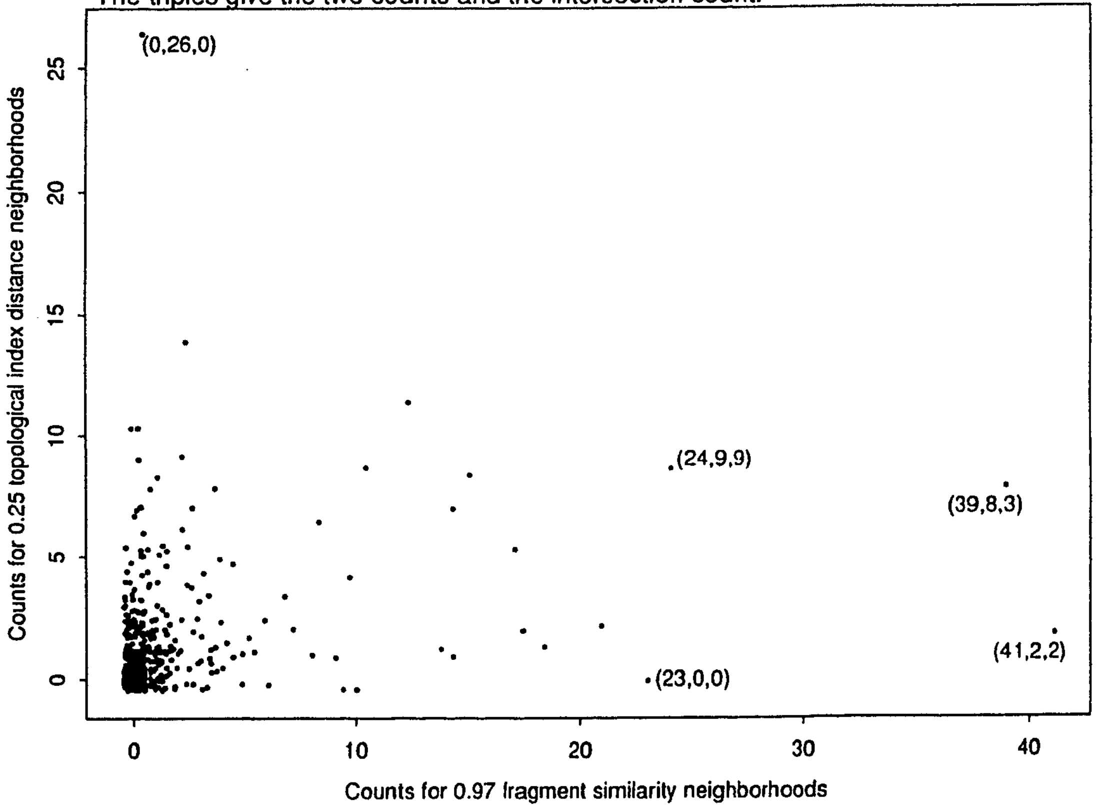
These two proximity measures are highly related although it may not be immediately apparent from the disparity in the forms of the information captured by their underlying vector representations. This relatedness becomes immediately apparent when one performs similarity searches using a common query structure. If the common query structure is a prostaglandin (a particular class of molecular structures), all of the most similar compounds in the databases by either proximity measure will be prostaglandins; if the common query structure is a benzodiazepine, all of the most similar compounds will be benzodiazepines, etc..However, such notions of relatedness between the two proximity measures presupposes an ability to define classes of compounds. Moreover, any basis of quantifying relatedness using these classes would reflect the idiosyncrases of the classification criteria.

Before illustrating the δ-dependence measure, it is informative to look at some counts employed in its computation. We selected 600 query structures at random from our database of over 100,000 structures. Both TI and FR similarity searches were performed for each query structure. For each query structure, we recorded the number of structures, excluding the query structure, in each similarity neighborhood (quadrat) as well as in the intersection of the neighborhoods. In this way, 600 3-tuples of counts were generated. All 600 similarity searches used a fixed cut-off value for the TI distance and another fixed cut-off value for the FR similarity. The experiment was then repeated for the same 600 query structures, but with different cutoff values for the two proximity measures. In the following discussion, only the results for a cut-off value of 0.25 for the TI distance and for a cutoff value of 0.97 for the FR similarity are presented.

Our first surprise was the complete lack of correlation seen in Figure 2 between the pairs of counts based on the TI and FR proximity measures. Since the 600 neighborhoods for each proximity measure share a common cut-off value or radius, one expects a high count to reflect a region (defined by the position in space of the query structure) with a relatively high value for the density function. Let f_{TI} and f_{FR} denote the density functions associated with how the structures are positioned in space under the TI and FR representations. Figure 2 suggests two things. First, for both densities, by far the largest proportion of density is associated with a very low density value, but occasionally one encounters an extremely dense region. Second, let $TI(z)$ and $FR(z)$ denote the TI and FR representations of structure z. Then the random variable $f_{TI}(TI(Z))$ has virtually no correlation with $f_{FR}(FR(Z))$ where Z denotes a randomly selected structure.

At first, this second finding totally surprised us. However, the apparent lack of correlation between the random variables $f_{TI}(TI(Z))$ and $f_{FR}(FR(Z))$ does not imply a lack of correlation between $TI(Z))$ and $FR(Z)$. To see this, imagine a transformation that differentially stretches a space on which a density function is defined without seriously altering neighboring relationships. Such a transformation would preserve the contiguous positioning of structures within a structural class by both proximity measures while at the same time allowing the two proximity measures to differ in how they "stretched out" the regions defining each structural class.

Figure 2. (-.45,.45)-Jittered plot of 600 count pairs using two different proximity measures. The triples give the two counts and the intersection count.



Although Figure 2 does not suggest any correlation between the low and high dense regions under TI distance with the low and high dense regions under FR similarity, it is not difficult to establish that their neighboring relationships are related using the counts of the number of structures in intersections of particular pairs of neighborhoods. For example, the point in Figure 2 with coordinates (24,9) corresponds to a pair of neighborhoods whose intersection contains 9 structures, i.e. the FR-similarity neighborhood is a subset of the TI-distance neighborhood. Suppose that the structures are distributed in "FR space" independently of their distribution in "TI space". If we had 100,000 structures in the database, the probability a randomly selected structure would fall in this TI neighborhood is 25/100,000 and the corressponding probability for the FR neighborhood is 9/100,000. If these two events are independent, the probability of a randomly selected structure falling in the intersection is the product of these two probabilities. It follows that the expected number of counts associated with two randomly selected neighborhoods of this size is roughly estimated by $25 \times 9/100,000 \cong 0.00225$. One can view the intersection counts as a Poisson random variable with mean 0.00225. Thus, our seeing an intersection count of 9 is extremely improbable under the assumption that the random variables $f_{TI}(TI(Z))$ and $f_{FR}(FR(Z))$ are independent.

Although interesting, this particular test does not provide a calibrated measure of dependence. For that we turn to the δ coefficient calibrated in Figure 1. With f_{TI} and f_{FR} playing the roles of f_1 and f_2 in the preceding section, we obtain the estimates and confidence intervals for the RACs given in Table 1. A sense for the histogram of the counts making up the two self-aggregation coefficients can be obtained from Figure 2. The distribution of the intersection counts for the joint density $f_{FR\times TI}$ is given by

count	0	1	2	3	4	5	6	9
freq	525	57	7	5	2	2	1	1

All 600 counts in which the product density was the design density were zeros. These were pooled with the preceding 600 intersection counts when estimating $\alpha(f_{FR\times TI}|g)$. The bootstrap confidence intervals were developed from 500 bootstrap samples from the sample quantile function of the observed counts.

Table 1

RAC	Estimate	95% CI
$\widehat{\alpha}(f_{FR})$	14.71	(12.3, 17.2)
$\widehat{\alpha}(f_{TI})$	3.95	(3.65, 4.25)
$\widehat{\alpha}(f_{FR\times TI})$	8.12	(5.7, 10.5)
$\widehat{\alpha}(f_{FR\times TI}|g)$	16.2	(11.3, 21.2)
$\widehat{\delta}(f_{FR\times TI})$	2.04	(1.31, 2.94)

It is easily shown that if $f_{FR\times TI} = f_{FR} \times f_{TI}$, then $\alpha(f_{FR\times TI}) = \alpha(f_{FR}) \times \alpha(f_{TI})$. Clearly, this is not the case, although we are still unsure of the meaning and significance of the fact that $\alpha(f_{FR\times TI})$ is so much less than the product of the self-aggregation coefficients. However, $\alpha(f_{FR\times TI}|g)$ is twice that of $\alpha(f_{FR\times TI})$, giving an estimate of two for $\delta(FR(Z), TI(Z))$. Based on the calibration of Figure 1, there is an extreme dependence $FR(Z)$ and $TI(Z)$.

References

Basak, S.C. Magnuson, V.R. Niemi, G.J. and Regal, R.R. (1988). Determining Structural Similarity of Chemicals Using Graph-Theoretic Indices, *Discrete Appl. Math.*, **19** 17-44.

Cheng, C. and Johnson, M. A. (1994a) Relative aggregation coefficients for describing and comparing densities in proximity spaces. manuscript.

Cheng, C. and Johnson, M. A. (1994b) Relative aggregation coefficient characterizations: a basis for inference on proximity data using random quadrat sampling. manuscript.

Cheng, C. and Johnson, M. (1994c). Relative Aggregation and Random Quadrat Sampling. *Computing Science and Statistics-Proceedings of the 26th Symposium on the Interface* (Editor, J. Sall) – this volume.

Johnson, M. A. (1989). A review and examination of the mathematical spaces underlying molecular similarity analysis. *Journal of Mathematical Chemistry* **3** 117-145.

Johnson, M. A. and Maggiora, G. M. (1990). *Concepts and Applications of Molecular Similarity*. New York, Wiley Inter-Science.

Willett, P (1987). *Similarity and Clustering in Chemical Information Systems*. Research Studies Press Ltd., Letchworth.

ROBUST EMPIRICAL AND HIERARCHICAL BAYES ESTIMATION OF NORMAL MEANS AND RATES IN LONGITUDINAL STUDIES

Ming Tan
Department of Biostatistics and Epidemiology
The Cleveland Clinic Foundation, Cleveland, OH 44195

Jean-François Angers
Département de mathématiques et de statistique
Université de Montréal, Montréal, H3C 3J7

Abstract

Robust empirical and hierarchical Bayes estimators for exchangeable normal means with heterogeneous variances are developed. The robust empirical Bayes estimator is obtained by using robust (or hierarchical) priors and the Newton-Ralphson algorithm. The robust hierarchical Bayes estimator is developed through t (particularly the Cauchy) priors with the computation being performed through the Gibbs sampler. It is shown that such robust estimators preserve the gain of shrinkage in the presence of extreme individual component estimators. Efron and Morris's classic example of estimating the toxoplasmosis prevalence rates is reconsidered. The method is then applied to the estimation of rates of change in longitudinal studies and is illustrated with an example. Further, the estimators are compared with those obtained via BLUP estimators of the random effects in SAS PROC MIXED through a simple random coefficient growth curve model.

1 INTRODUCTION

With the recent development of computational tools such as the Gibbs sampler, complex data can be analyzed through a comprehensive Bayesian hierarchical model. In this paper, however, we consider some important estimation properties in the basic model of estimating exchangeable normal means (or random-effects), such as the the estimator's robustness with respect to prior misspecifications and outlying observations. Such estimation is often needed in practice, as is demonstrated in Morris (1983), Breslow (1990) and Louis (1991), and can be summarized as estimating $\beta_1,\ldots, \beta_k$ simultaneously starting with their independent unbiased estimators $b_1,\ldots, b_k$. Often it is assumed that $b_i \mid \beta_i \sim N(\beta_i, d_i^2)$, $i = 1,\ldots, k$, independently. It is now well known that shrinkage estimators (Morris, 1983) can generally improve upon the usual maximum likelihood estimator (b_i) for β_i in terms of achieving smaller squared error risk. And the shrinkage estimators are often derived from a Bayes approach by assuming that the β_i's are from a certain probabilistic distribution. This method has been shown to be useful in problems where the scientific objectives were not directly one of simultaneous estimation, e.g., it provides a way to correct for the effect of regression to the mean and gives estimators of regression coefficients which yield uniformly smaller prediction mean square error in linear and logistic regression (Copas, 1983); and it also gives estimators with uniformly smaller variances in discrete event simulation with control variates (Tan and Gleser, 1992) and estimators of common odds ratio using concordant pairs (Liang and Zeger, 1988).

In the simpliest case when $d_i^2 = \sigma^2$ for all $i = 1,\ldots, k$, $b_i \mid \beta_i \sim N(\beta_i, \sigma^2)$ with a conjugate (Gaussian) prior $\beta_i \sim N(\beta, A)$, the shrinkage estimator of β_i for $i = 1,\ldots, k$ as proposed in Morris (1983) is of the form

$$\hat{\beta}_i = b_i - \min\left(\frac{k-3}{k-1}, \frac{(k-3)\sigma^2}{\sum_{i=1}^k (b_i - \bar{b})^2}\right)(b_i - \bar{b}), \quad (1.1)$$

where $\bar{b}$ is the grand mean of the b_i's. This estimator has smaller squared error risk than the maximum likelihood estimator, provided that $k \geq 4$. When the variances d_i^2 are not equal, an iterative algorithm is needed to calculate the empirical Bayes estimator. Tan and Gleser (1992) have studied the magnitude of potential improvement of these estimators. The gain would be substantial if the individual means are reasonably similar.

However, the conjugate priors are not necessarilly robust (with respect to possible misspecifications of priors). In fact, as pointed out in Berger (1985, Chapter 4), when the likelihood function is concentrated in the tail of the prior distribution, conjugate priors should probably be avoided. Although the usual normal conjugate prior for estimating the normal means is robust within the class of all prior distributions with finite first two moments (Morris, 1983), the moments depend on the tail of the distribution and are thus highly variable. For instance, two priors may be virtually indistinguishable but may have quite different moments (Berger, 1985), and some highly robust priors (such as the Cauchy priors) do not have moments. Sometimes the estimator using the conjugate Gaussian prior (such as 1.1) has been referred to as being robust in a conservative sense in that if the prior is fully wrong or if one b_i is outlying (thus in violation of exchangeability), the empirical Bayes (EB) estimates would collapse back to the usual maximum likelihhod estimators, resulting in no harm but nullifying the potential gain. Therefore estimators that preserve the gain of shrinkage in the presence of outlying components are very appealing because the rest of the components (the individual β_i's) can still benefit from borrowing strength from the ensemble. This refined robustness can be achieved by using flat-tailed priors or by hierarchical modeling (Berger, 1985, Angers and Berger, 1991). When the variances are not equal, the first stage parameter estimators in the hierarchical model can be derived from Angers (1992) when the degrees of freedom of the t-prior is odd. In general when the variances are heterogeneous, analytic solutions with t-priors seem extremely difficult to obtain. With the computation being performed via Gibbs sampler, such robust estimates of random effects can be easily extended to the general mixed-effects model of Laird and Ware (1982).

The purpose of this paper is to develop empirical and hierarchical Bayes estimators with the refined robustness. The class of t-priors (the Cauchy prior in particular) is used to obtain the robust heirarchical estimators with computation being performed using the Gibbs sampler (Geman and Geman, 1984, Gelfand et al., 1990).

As a quicker alternative, we first use the robust prior in Berger (1985) to derive robust empirical Bayes estimators through use of the Newton-Ralphson algorithm in §2.1. Hierarchical Bayes modeling via the Gibbs sampler is considered in §2.2. A data set from the literature (Efron and Morris, 1975) is reconsidered in §2.3. In §3, the method is applied to longitudinal studies where estimation of the rates of individual change is of interest and is illustrated with a real life example. The paper is concluded with a discussion in §4.

2 ROBUST ESTIMATES

2.1 Robust empirical Bayes estimates

The robust prior developed in Berger (1985) is based on the consideration of the admissibility of the Bayes estimators (Strawderman and Cohen, 1971). Using this prior, the model can be given as

$$b_i \sim N(\beta_i, d_i^2), \text{ and } \beta_i \sim N(\mu, B(\lambda_i)), \tag{2.1}$$

where $B(\lambda_i) = (d_i^2 + A)/(2\lambda_i) - d_i^2$, and λ_i has density $\pi(\lambda_i) = 0.5\sqrt{\lambda_i} I_{(0,1)}(\lambda_i)$. Given μ and A, the posterior mean and variance of β_i are:

$$E(\beta_i|b) = b_i - \frac{2d_i^2}{d_i^2 + A}\left(\frac{1}{\|b_i\|^2} - \frac{1}{e^{\|b_i\|^2} - 1}\right)(b_i - \mu),$$

$$\begin{aligned} var(\beta_i|b) &= d_i^2 - \frac{2d_i^4}{d_i^2 + A} \\ &\times \left[\frac{1}{e^{\|b_i\|^2} - 1}\left(\frac{2\|b_i\|^2}{1 - e^{-\|b_i\|^2}} - 1\right) - \frac{1}{\|b_i\|^2}\right], \end{aligned} \tag{2.2}$$

where $\|b_i\|^2 = (b_i - \mu)^2/(d_i^2 + A)$. The marginal distribution of b_i is

$$m(b_i|\mu, A) = \frac{1}{2\sqrt{\pi}}\frac{1}{\sqrt{d_i^2 + A}}\frac{1 - e^{-\|b_i\|^2}}{\|b_i\|^2}.$$

Parameters μ and A can be estimated using the maximum likelihood method via the Newton-Ralphson algorithm.

Another advantage of the above estimator is that it easily yields subjective hierarchical Bayes estimates (Berger and Robert, 1990) for any plausible μ and A. However this prior should be used with caution. It may cause the estimator to collapse back to b_i when A/d_i^2 is too big. In other words, the prior may be so flat such that its effect on the estimators is essentially the same as that of a uniform (noninformative) prior.

2.2 Robust Hierarchical Bayes Estimate

The robust hierarchical Bayes estimate has many advantages over the empirical Bayes estimate (Berger and Robert, 1990). A main advantage is that it takes into account the error due to the estimation of the hyperparameters, whereas the empirical Bayes method ignores such error. Another advantage is that in the hierarchical model the marginal posterior distributions can be estimated via Gibbs sampling (Gelfand et al, 1990). Thus, standard errors and confidence intervals can be developed easily.

As shown in Berger (1985, pages 195-196), a Cauchy prior is more reasonable in terms of the posterior robustness and Bayesian risk if we are uncertain as to which priors best describe our prior belief. We now consider the following hierarchical model

$$b_i|\beta_i \sim N(\beta_i, d_i^2), \quad \text{and } \beta_i|\mu, \sigma^2 \sim t_1(\mu, \sigma^2, v_0)$$

$$\mu \sim N(\eta, C), \quad \sigma^2 \sim Gamma(p, q), \qquad (2.3)$$

where the (multivariate) t-distribution, with location parameter μ and scale matrix $\sigma^2 I$ and dimension k, denoted by $u \sim t_k(\mu, \sigma^2 I, v_0)$, has density of the form:

$$f_k(u|\mu, \sigma^2, v_0) = \frac{g(v_0)}{\sigma^k} \frac{1}{(v_0 + \frac{(u-\mu)'(u-\mu)}{\sigma^2})^{\frac{k+v_0}{2}}},$$

where $v_0, \sigma^2 > 0, g(v_0) = const.$ Of particular interest are the two special cases: 1) if $v_0 = 1$, $k = 1$, then $f_1(u|\mu, \sigma^2)$ is the Cauchy prior with median μ and quartiles $\mu \pm A$; 2) and if $v_0 = \infty$, $f_k(u|\mu, \sigma^2) = N_k(\mu, \sigma^2 I_p)$, is the Gaussian prior. Since the t-distribution is a mixture of the Gaussian and inverse gamma distributions, all conditional distributions used in the Gibbs sampling have closed forms and thus the algorithm is very efficient. In fact, the t-distribution can be decomposed into

$$u|\tau^2 \sim N_p(\mu, \tau^2 I), \text{ and } \tau^2 \sim IG(\frac{v_0}{2}, \frac{v_0\sigma^2}{2}),$$

where

$$IG(v_0/2, u_0/2) = (u_0/2)^{v_0/2} e^{-u_0/2v} v^{-(v_0/2+1)} \Gamma^{-1}(v_0/2)$$

is the density function of the inverse Gamma distribution. So all the conditional distributions are given as follows:

$$[\beta_i|\tau^2, \mu, \sigma^2, (b_i)] \sim N(\frac{\tau^2}{d_i^2 + \tau^2} b_i + \frac{d_i^2}{d_i^2 + \tau^2} \mu, \frac{d_i^2 \tau^2}{d_i^2 + \tau^2}),$$

$$\begin{aligned}
[\tau^2|(\beta_i), \mu, \sigma^2, (b_i)] &\sim IG(\frac{v_0 + k}{2}, \frac{v_0\sigma^2}{2} + \frac{|\beta - \mu|^2}{2}), \\
[\mu|(\beta_i), \tau^2, \sigma^2, (b_i)] &\sim N(\frac{C}{k\tau^2 + C}\bar{\beta} + \frac{k\tau^2}{k\tau^2 + C}\eta, \\
&\qquad \frac{C\tau^2}{k\tau^2 + C}), \\
[\sigma^2|(\beta_i), \tau^2, \mu, (b_i)] &\sim Gamma(p, \frac{1}{(2\tau^2)^{-1} + q^{-1}}).
\end{aligned}$$

Then the Gibbs sampling can be applied to the hierarchical model specified in (2.1). Given the data (b_i), one can obtain the needed marginal distribution (say $\pi(\beta_i|b_i)$) from the Gibbs sampling.

A comparison between empirical and hierarchical Bayes estimators is given in Kass and Steffey (1989) in which approximations of the posterior variances are also given. Applying the approximation to the model given by equations (2.1) and (2.3), one can see that the additional term needed to take into account the estimation of the hyperparameters μ and A in (2.1) and μ and σ in (2.3) is of order $O(1/[n_i^2 k])$ while the main term is of order $O(n_i^{-1})$. Consequently, equation (2.2) or the conditional variance based on $\pi(\beta_i|data, \hat{\lambda})$ is a good approximation of the posterior variance when n_i is relatively small and k is large. In this case the empirical Bayes estimators can serve as an adequate approximation to those obtained from the hierarchical model. However, the robust hierarchical Bayes model gives estimates which are resistant to both misspecification of the prior and outlying component estimates, as mentioned earlier.

2.3 Estimating toxoplasmosis prevalence rates

We now consider an example taken from Efron and Morris (1975), in which the prevalence rates of toxoplasmosis in 36 El Salvadorian cities were estimated. The prevalence rates in Table 1 are standardized and the variances are known from the binomial distribution, and differ because of unequal number of patients sampled in different cities. Table 1 gives the robust empirical and hierarchical Bayes estimates, as well as the empirical Bayes estimates developed in their paper as a comparison. The maximum likelihood estimate via the Newton-Ralphson algorithm converged at $\mu = 0.024$, and $A = 3.26$ after 55 iterations starting from the mean and variance of the 36 prevalence rates. The Gibbs sampling algorithm converged roughly with 160 cycles of $m = 50$ drawings in that there was little change in the successive posterior distributions thereafter at 200, 240 cycles. In fact, the change in quartiles of the posterior distributions was less than 10^{-6}. The initial values were $\eta_0 = -0.0419$, $C_0 = 12$, $p_0 = 0.2$, and $q_0 = 0.001$, indicating rather vague prior knowledge about these parameters. It seems in this example that the normal prior is indeed quite robust, as the robust hierarchical Bayes estimators and Efron and Morris's empirical Bayes estimators are very similar except for only a few cities in which the prevalence rates are more at the extremes. This similarity is what is expected of the (refined) robust hierarchical estimators. The robust empirical Bayes estimates, however, are essentially the same as the original estimated prevalence rates. It is probably too conservative in that the information between cities did not add any new information about the prevalence rates. In fact, $\min(A/d_i^2) = 655$ is quite large in this case.

3 RATES IN LONGITUDINAL STUDIES

Often in longitudinal studies the rate of change of the response over time is of primary interest, and such change is often approximately linear (possibly after some transformations, and/or over a short period of follow-up). For instance, the decline of lung and renal functions is linear for certain patient populations. In this case, it is reasonable to reduce the data to slopes and their standard deviations by linear regression for each individual (Hui and Berger, 1983). Because the subjects under study share some common characteristics (belonging to a certain population), it is reasonable to assume that their individual rates of change come from a common probability distribution. Consequently, shrinkage estimators of the individual rates are desirable (Morris, 1983). This approach ignores the intercept and thus loses some information in comparison with the the Gaussian random effects model (Laird and Ware, 1982) or more generally a repeated measures model of Jennrich and Schluchter(1986) which allows the modelling of various within-subject correlation structures.

We now consider a prospective study in ophthalmology where intraocular gas was used in complex retinal surgeries to provide internal tamponade of retinal breaks in the eye. An important issue was to estimate the kinetics (e.g., decay rate, half-life, and so on) of the disappearance of the gas. After gas was injected into their eyes, 31 patients were seen three to eight (average of 5) times over a three-month period, and the volume of the gas in their eyes was recorded.

Let y_{ij} be the j^{th} gas volume for the i^{th} individual at day x_{ij}. Some initial analysis suggested that the volume (in percent) of the intraocular expansile gas (C_3F_8) decreases slowly in the first few days after maximal expansion, then it decreases more rapidly and finally more slowly (pródducing an S-shaped curve). Thus a logit transformation was first made on the gas volume:

$$z_{ij} = \log\left(\frac{y_{ij}+0.05}{1-y_{ij}+0.05}\right),$$

where 0.05 was added to avoid zero denominators. Then a linear model can be assumed:

$$z_{ij} = \alpha_i + \beta_i x_{ij} + e_{ij},$$

where e_{ij} is the Gaussian error term with mean 0 and variance σ_i^2. Separate linear regressions using each subject's data are used to obtain the quantities:

$$b_i \sim N(\beta_i, d_i^2), \quad d_i^2 = \frac{\sigma_i^2}{\sum_{i=1}^{n_i}(x_{ij}-\bar{x}_i)^2} \qquad (3.1)$$

$$\text{and} \quad s_i^2 \sim \sigma_i^2\chi^2_{n_i-2},$$

where d_i^2 is the usual variance estimate of the slope b_i and σ_i^2 is estimated from the s_i^2. Hui and Berger (1983) also use the empirical Bayes estimates of σ_i^2's as a compromise between $s_i^2/(n_i-2)$, the individual estimate, and $\Sigma s_i^2/\Sigma n_i$, the pooled estimate. Both estimates are independent of the b_i's. However, we only use the individual estimates of σ_i^2 to illustrate the method.

The goal is to find an improved estimator for each individual decay rate β_i to get a better idea of the variability of these rates. Robustness considerations are particularly relevant here because previous studies suggested the gas decay rate was highly variable (Meyers et al., 1992). A Cauchy prior with median μ and quantiles $\mu \pm \sigma$ seems to be plausible. Further, a normal hyperprior on μ and a gamma prior on σ^2 is used. The initial values were given by $\eta_0 = -0.08$, $C_0 = 12$, $p_0 = 0.2$, $q_0 = 0.0001$ indicating a rather vague prior knowledge about these parameters was assumed. Different starting values were used for different cycles (iterations). The convergence was achieved roughly with 240 cycles of $m = 40$ drawings in that there was little change in the histograms of the posterior distributions thereafter at $240, 300, 360, 400, 420$ cycles. The changes in quartiles were less than 10^{-6}. The decay rates were estimated based on the data after 420 cycles of iterations. The robust hierarchical model gives improved estimators of the decay rates and their standard errors by borrowing strength from the ensemble and thus provides a more accurate picture of the variation of the individual gas decay rates. This can be more clearly shown by looking at the plot of these rates over the cases (not shown here). Table 2 gives the least square slopes, RHB estimates and their standard errors and a 90 % confidence interval for each individual decay rate.

In this data set, the decay rate for case 30 is outlying, being beyond 1.5 times the interquartile range. We have found that the RHB estimates are quite close to those obtained when the outlier is removed (see Table 2). Thus our estimate is indeed quite robust with respect to outlying rates.

Finally we fitted a random coefficient growth curve model. The estimated ***best linear unbiased predictors*** (BLUPs) of the individual rates of decline are obtained using SAS PROC MIXED. Since these estimators are in fact shrinkage estimators of the slopes using normal priors, they may not have the refined robustness (with respect to outlying individual slopes) and could give BLUP estimators which are more or less the same as the original least square slopes. Thus the possible gain of using the random effects model is diminished. This indeed appears to be the case (see Table 2).

Table 1

Estimates of Toxoplasmosis Prevalence Rates in 36 El Salvadorian Cities

City	b_i	d_i	EB	RHB	REB
1	0.293	0.304	0.035	0.0776	0.2861
2	0.214	0.039	0.192	0.1962	0.2139
3	0.185	0.047	0.159	0.1673	0.1849
4	0.152	0.115	0.075	0.0764	0.1515
5	0.139	0.081	0.092	0.0924	0.1388
6	0.128	0.061	0.100	0.0878	0.1279
7	0.113	0.061	0.088	0.0910	0.1129
8	0.098	0.087	0.062	0.0461	0.0978
9	0.093	0.049	0.079	0.0844	0.0930
10	0.079	0.041	0.070	0.0688	0.0790
11	0.063	0.071	0.045	0.0599	0.0629
12	0.052	0.048	0.044	0.0377	0.0520
13	0.035	0.056	0.028	0.0267	0.0350
14	0.027	0.040	0.024	0.0227	0.0270
15	0.024	0.049	0.020	0.0010	0.0240
16	0.024	0.039	0.022	0.0289	0.0240
17	0.014	0.043	0.012	0.0096	0.0140
18	0.004	0.085	0.003	-0.0005	0.0040
19	-0.016	0.128	-0.007	-0.0168	-0.0158
20	-0.028	0.091	-0.017	-0.0194	-0.0279
21	-0.034	0.073	-0.024	-0.0440	-0.0339
22	-0.040	0.049	-0.034	-0.0252	0.0400
23	-0.055	0.058	-0.044	-0.0484	-0.0549
24	-0.083	0.070	-0.060	-0.0521	-0.0829
25	-0.098	0.068	-0.072	-0.0640	-0.0978
26	-0.100	0.049	-0.085	-0.1025	-0.0999
27	-0.112	0.059	-0.089	-0.0979	-0.1119
28	-0.138	0.063	-0.106	-0.1370	-0.1378
29	-0.156	0.077	-0.107	-0.1190	-0.1557
30	-0.169	0.073	-0.120	-0.1436	-0.1687
31	-0.241	0.106	-0.128	-0.2131	-0.2402
32	-0.294	0.179	-0.083	-0.1575	-0.2911
33	-0.296	0.064	-0.225	-0.2527	-0.2956
34	-0.324	0.152	-0.114	-0.2125	-0.3217
36	-0.397	0.158	-0.133	-0.2079	-0.3940
36	-0.665	0.216	-0.140	-0.3115	-0.6560

b_i=original prevalence estimates; REB=robust empirical Bayes estimates;
RHB=robust hierarchical Bayes estimates;
EB=Empirical Bayes estimates from Efron and Morris (1975).

Table 2

Estimates and Standard Errors for Intraocular Gas Decay Rates Over 3 Months

Case No.	Decay Rate b_i	PROC MIXED	RHB	Standard Error d_i	Lower Limit	Upper Limit	RHB without Outlier
1	-0.079192	-0.07964	-0.085599	0.054523	-0.175281	0.004083	-0.081735
2	-0.113267	-0.10898	-0.095713	0.043920	-0.167956	-0.023470	-0.091279
3	-0.097290	-0.09466	-0.089884	0.040208	-0.156020	-0.023747	-0.087211
4	-0.107540	-0.10198	-0.094236	0.041252	-0.162090	-0.026382	-0.091046
5	-0.058366	-0.07382	-0.076427	0.039428	-0.141280	-0.011574	-0.072782
6	-0.108757	-0.09548	-0.090590	0.055249	-0.181467	0.000287	-0.087030
7	-0.052102	-0.05499	-0.073855	0.038536	-0.137241	-0.010469	-0.070653
8	-0.105155	-0.10054	-0.091394	0.050133	-0.173856	-0.008932	-0.087452
9	-0.067717	-0.07468	-0.082851	0.048989	-0.163430	-0.002272	-0.078637
10	-0.049678	-0.05832	-0.060981	0.018936	-0.092128	-0.029834	-0.058885
11	-0.058768	-0.06361	-0.071113	0.027062	-0.115625	-0.026601	-0.068502
12	-0.016401	-0.05167	-0.070124	0.049017	-0.150750	0.010503	-0.066345
13	-0.065946	-0.07488	-0.083424	0.053049	-0.170681	0.003833	-0.080303
14	-0.057907	-0.06795	-0.073214	0.033444	-0.128225	-0.018204	-0.070318
15	-0.049867	-0.06766	-0.071486	0.035762	-0.130309	-0.012662	-0.067842
16	-0.068265	-0.07027	-0.080510	0.038218	-0.143372	-0.017647	-0.076105
17	-0.057945	-0.06554	-0.079484	0.049478	-0.160868	0.001900	-0.076851
18	-0.082468	-0.08240	-0.084660	0.025244	-0.126183	-0.043136	-0.081232
19	-0.049593	-0.05297	-0.051792	0.004164	-0.058641	-0.044943	-0.051194
20	-0.046228	-0.05408	-0.058733	0.019698	-0.091134	-0.026332	-0.057109
21	-0.139922	-0.12264	-0.103016	0.045781	-0.178319	-0.027712	-0.098716
22	-0.118010	-0.10224	-0.096735	0.044064	-0.169214	-0.024257	-0.092347
23	-0.102168	-0.09877	-0.093856	0.034720	-0.150965	-0.036746	-0.089928
24	-0.159127	-0.14065	-0.105545	0.049775	-0.187418	-0.023672	-0.101160
25	-0.152269	-0.11091	-0.096112	0.058156	-0.191771	-0.000454	-0.091959
26	-0.072401	-0.07264	-0.081338	0.037129	-0.142410	-0.020266	-0.077196
27	-0.077489	-0.07863	-0.081373	0.026484	-0.124935	-0.037810	-0.078726
28	-0.120120	-0.10383	-0.112152	0.014613	-0.136189	-0.088115	-0.110226
29	-0.091313	-0.08975	-0.087575	0.063681	-0.192320	0.017170	-0.098504
30	-0.314234	-0.10813	-0.090982	0.070123	-0.206324	0.024360	*
31	-0.085229	-0.08505	-0.084933	0.021638	-0.120525	-0.049341	-0.083128

*Outlying case #30 was removed.
RHB = robust hierarchical Bayes estimate

4 DISCUSSION

In summary, we used hierarchical modeling (through use of a Cauchy prior in particular) to obtain estimators of normal means (or random effects) which are robust with respect to prior misspecifications and outlying individual means. Thus the gain of shrinkage is preserved. It is worth pointing out that the same problem with Gaussian priors persists in the linear mixed-effects models of Laird and Ware (1982) for analyzing longitudinal data. The effect of assuming a Gaussian random effect is that the potential advantage of a random-effects model may vanish simply because of one outlying individual's random effects. It is however easy to incorporate the robust priors studied in this paper into these models if the computation is performed using the Gibbs sampler as is in Gilks et al.(1993).

ACKNOWLEDGEMENT

The authors thank James Berger for his valuable comments regarding the priors and Mark Schluchter and Michael Kutner for their helpful suggestions. They also thank Dr. S.M. Meyers for making his data available for inclusion as an example.

References

[1] Angers, J.F. and Berger, J.O. (1991). Robust hierarchical Bayes estimation of exchangeable means. *The Canadian Journal of Statistics*, **19**, 39–56.

[2] Angers, J.F. (1992). Use of student-t prior for the estimation of normal means: a computational approach. In *Bayesian Statistics IV*, J. M. Bernardo *et al.* (Eds.), Oxford University Press.

[3] Berger, J.O. (1985). *Statistical Decision Theory and Bayesian Analysis.* New York: Springer-Verlag.

[4] Berger, J.O. and Robert, C. (1990). Subjective hierarchical Bayes estimation of a multivariate normal mean: on the frequentist interface. *Ann Stat* **18**, 617–651.

[5] Breslow, N.(1990). Biostatistics and Bayes. *Stat Science*, 5:267–295.

[6] Copas, J.B. (1983). Regression, prediction and shrinkage. *J R Stat Soc B*, **45**, 311–354.

[7] Efron, B. and Morris, C. (1975). Data analysis using Stein's estimator and its generalizations. *J Am Stat Assoc* **70**, 311–319.

[8] Gelfand, A.E., Hills, S.E., Racine-Poon A., and Smith A.F.M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *J Am Stat Assoc* **85**, 972–985.

[9] Geman, S., and Geman, D. (1984). Stochastic ralaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans Pattern Analysis and Machine Intelligence*, **6**, 721–741.

[10] Gilks, W.R., Wang, C.C., Yvonnet, B. and Coursaget, P. (1993). Random-effects models for longitudinal data using Gibbs sampling. *Biometrics*, **49**, 441–453.

[11] Hui, S. and Berger, J.O. (1983). Empirical Bayes estimation of rates in longitudinal studies. *J Am Stat Assoc* **78**, 753–759.

[12] Jennrich, R.I. and Schluchter, M.D. (1986). Unbalanced repeated measures models with structured covariance matrices. *Biometrics*, **42**, 805–820.

[13] Kass, R.E. and Steffey, D. (1989). Approximate Bayesian inference in conditionally independent hierarchical models (Parametric Bayes methods). *J Am Stat Assoc* **84**, 717–726.

[14] Laird, N.M. and Ware, J.H. (1982). Random effects models for longitudinal data. *Biometrics* **38**, 963–974.

[15] Liang, K.Y. and Zeger, S.L. (1988). On the use of concordant pairs in matched case-control studied. *Biometrics*, 44, 1145–1156.

[16] Louis, T. (1991). Using empirical Bayes methods in biopharmaceutical research. *Statistics in Medicine* **10**, 811–829.

[17] Meyers, S.M., Ambler, J.S., Tan, M., Werner J.C. and Suber S.H. (1992). Variation of perfluoropropane disappearance after vitrectomy. *RETINA* **12**, 359-363.

[18] Morris, C. (1983). Parametric empirical Bayes inference: theory and applications. *J Am Stat Assoc*, **78**, 47–65.

[19] Strawderman, W.E. and Cohen, A. (1971). Admisibility of estimators of the mean vector of a multivariate normal distribution with quadratic loss. *Ann Math Stat*, **42**, 270–296.

[20] Tan, M. and Gleser, L. (1992). Minimax estimators for location vectors in elliptical distributions with unknown scale parameter and its application to variance reduction in simulation. *Ann Institute Stat Math*, 44, 537–550.

Recursive Partitioning and Event Rate Data

Terry M. Therneau
Mayo Foundation
Rochester, Minnesota 55905

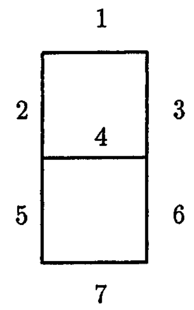
Abstract

Recursive partitioning methods, also known are tree or CART models, have been applied to several kinds of data, including the cases where the response y is a continuous variable, a category or class, a survival time, and a longitudinal response pattern. In this work we extend the methods to the prediction of an observed response rate (number of events)/(time observed). The building and ordering of a tree model work well, but there are some open issues in cross-validation of the final model. Finally, some connections are noted to other work on trees for survival data.

1 Introduction

Recursive partitioning is a method for growing binary decision trees, where each node or split represents a decision, e.g., go to the left if age < 40, and the terminal leaves give the predicted values. These methods date back to the AID (Automatic Interaction Detection) program developed by Morgan and Sonquist in the early 1960s, and received a strong theoretical boost with the CART (Classification and Regression Trees) work of Brieman, et.al. in the 1980s [1]. A famous example is the digit recognition problem.

Consider the segments of an unreliable digital readout where each light is correct with probability 0.9, e.g., if the true digit is a 2, the lights 1, 3, 4, 5, and 7 are on with probability 0.9 and lights 2 and 6 are on with probability 0.1. Construct test data where $Y \in \{0, 1, ..., 9\}$, each with proportion 1/10 and the $X_i, i = 1, \ldots, 7$ are i.i.d. bernoulli variables with parameter depending on Y. $X_8 - X_{24}$ are generated as i.i.d bernoulli $P\{X_i = 1\} = .5$, and are independent of Y. They correspond to embedding the readout in a larger rectangle of random lights. A sample of size 200 was generated accordingly and the CART procedure applied to build the tree. The results are shown in figure 1.



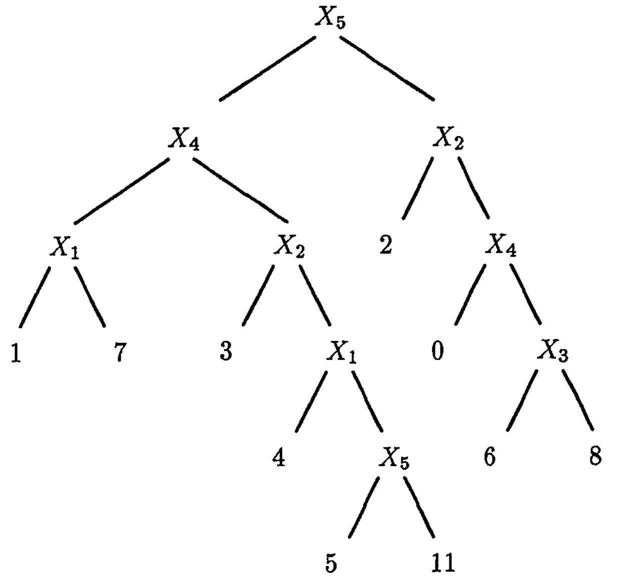


Figure 1: Optimally pruned tree for the stochastic digit recognition data

Tree methods have been applied to regression and classification problems [1], survival analysis [3], longitudinal analysis [6] and others. The goal of this research is to extend the methodology to event rate data. The model in this case is

$$\lambda = f(x)$$

where λ is an event rate and x is some set of predictors. As an example consider hip fracture rates. For each county in the United States we can obtain

- number of fractures in patients age 65 or greater (from Medicare files)
- population of the county (US census data)
- potential predictors such as
  - socio-economic indicators
  - number of days below freezing
  - ethnic mix
  - physicians/1000 population
  - etc.

Such data would usually be approached by using Poisson regression; can we find a tree based analogue?

2 Recursive partitioning ingredients

A tree based method has four main ingredients

1. A split criteria. This is used to determine the "best" available split of a node into two daughter nodes.
2. An impurity criteria. This is used to measure the "homogeneity" of a node, and is used to order the possible sub-trees (sub-models) of the full tree model.
3. Labeling: An "average response" for each node.
4. Prediction error: The error in prediction for a new observation, should it be predicted using this node. This is needed for cross-validation but not for building or ordering the tree.

For tree based regression, these are

1. the between groups sum of squares,
2. the within node sum of squares,
3. the mean and variance of a node,
4. $(y - \hat{y})^2$.

For tree based classification there are several variations. Choices include

1. One of
   - the likelihood ratio test for $H_0 : p_1 = p_2$, where p_1 and p_2 are the vector of proportions in the two daughter nodes.
   - the Gini criterion
   - the twoing criterion (see [1])
2. One of
   - the binomial deviance within the node
   - the risk of a node, based on priors and a loss matrix
3. The predicted class for the node, or the vector of class probabilities
4. One of
   - the prediction loss L(observed class, predicted class), where L is the loss matrix
   - the predicted contribution to the deviance.

(Many other choices have been explored for this problem).

In adding criteria for rates regression to this ensemble, the guiding principle was the following: the between groups sum-of-squares is not a very robust measure, yet tree based regression works very well. So do the simplest thing possible.

Let c_i be the observed event count for observation i, t_i be the observation time, and $x_{ij}, j = 1, \ldots, p$ be the predictors.

Labels: The observed event rate and the within-node deviance

$$\hat{\lambda} = \frac{\#\text{ events}}{\text{total time}} = \frac{\sum c_i}{\sum t_i}$$

$$D = \sum \left[c_i \log\left(\frac{c_i}{\hat{\lambda} t_i}\right) - (c_i - \hat{\lambda} t_i) \right]$$

Splitting rule: The likelihood ratio test for two Poisson groups

$$D_{\text{parent}} - \left(D_{\text{left son}} + D_{\text{right son}}\right)$$

Purity: The within node deviance.

Prediction: The deviance contribution for a new observation, using $\hat{\lambda}$ of the node as the predicted rate.

3 Improving the method

There is a problem with the criterion just proposed, however: cross-validation of a model often produces an infinite value for the deviance. The simplest case where this occurs is easy to understand. Assume that some terminal node of the tree has 20 subjects, but only 1 of the 20 has experienced any events. The cross-validated error (deviance) estimate for that node will be

$$\ldots + c_i \log(c_i / 0 * t_i) + \ldots$$

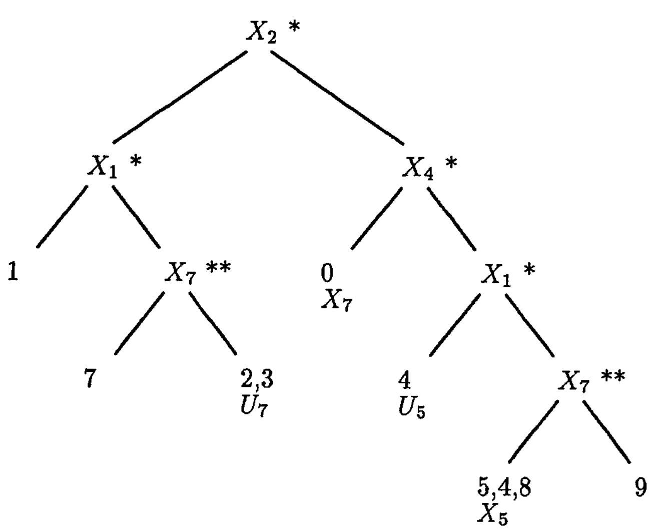


Figure 2: Rates recognition

which is infinite for $c_i > 0$. The problem is that when $\hat{\lambda} = 0$ the occurrence of an event is infinitely improbable, and, using the deviance measure, the corresponding model is infinitely bad.

One might expect this phenomenon to be fairly rare, but unfortunately it is not so. One given of tree-based modeling is that a right-sized model is arrived at by purposely overfitting the data and then pruning back the branches. A program that aborts due to a numeric exception during the first stage is embarrassing to say the least.

Of more concern is that this edge effect does not seem to be limited to the pathologic case detailed above. Any near approach to the boundary value $\lambda = 0$ leads to large values of the deviance, and the procedure tends to discourage any final node with a small number of events.

An ad hoc solution is to use the revised estimate

$$\hat{\hat{\lambda}} = \max\left(\hat{\lambda}, \frac{k}{\sum t_i}\right)$$

where k is 1/2 or 1/6. This is similar to the starting estimates used in the GLM program for a Poisson regression. This is unsatisfying, however, and we propose instead using a shrinkage estimate.

Assume that the true rates λ_j for the leaves of the tree are random values from a Gamma(μ, σ) distribution. Set μ to the observed overall event rate $\sum c_i / \sum t_i$, and let the user choose as a prior the coefficient of variation $k = \sigma/\mu$. A value of $k = 0$ represents extreme pessimism ("the leaf nodes will all give the same result"), whereas $k = \infty$ represents extreme optimism. The Bayes estimate of the event rate for a node works out to be

$$\hat{\lambda}_k = \frac{\alpha + \sum c_i}{\beta + \sum t_i},$$

where $\alpha = 1/k^2$ and $\beta = \alpha/\hat{\lambda}$.

This estimate is scale invariant, has a simple interpretation, and shrinks least those nodes with a large amount of information. In practice, a value of $k = 10$ does essentially no shrinkage. All tests were done with $k = 1$.

4 Examples

As an example, we consider a variant of the digit recognition problem. Let X_1 to X_7 be the segments of a digital readout, as in the earlier example, where each segment is in error 20% of the time. Let U_1 to U_{10} and B_1 to B_{10} be extraneous predictors with uniform(0,1) and binomial(.5) distributions, respectively. The true class of the observations is evenly divided over the digits 0–9, but the true class is not observed. Instead we observe a Poisson count with rate $\lambda = .34$ for class 0 and rate $\lambda = 3.4$ for class 9, the true rates are evenly spaced on a logarithmic scale. The number of observations and the total time on test was varied between simulations.

A typical tree for $n = 1000$ and $t_i \sim U(.5, 1.5)$ is shown in figure 2. With this choice for n and t there were on average 1000 events, which is a fairly large sample.

Each internal node of the tree is labeled with the variable used to split at that node. The nodes marked with a double asterisk are retained if one uses the minimum cross-validated error rule, and those with an asterisk are retained if the "1 SE" rule is used. Each leaf is labeled with the class(es) that would be routed to that leaf if X_i were measured without error; for some of the leaves we also show the next variable that was chosen by the splitting rule (although the split was not retained). In ten independent runs of this simulation, the same qualitative results were obtained.

First, this is a hard problem. A plot (not shown) of the observed event rates c_i/t_i versus the class shows considerable overlap. Classes 1–3 were never well resolved, and the high error rate for the true predictors makes deep trees difficult for this sample size.

Secondly, even with shrinkage the cross-validation criteria seems to recommend trees that are too small. The 'best' tree, i.e., the one with lowest cross-validation error, sometimes missed informative splits, such as the split on X_5 at the bottom of figure 2 (it also sometimes included an uninformative split). The '1 SE' rule, however, consistently trimmed off 1-2 informative splits from the best tree.

Third, the method is asymptotically consistent.

When the average time of observation t_i was increased to 10, keeping the same event rates (so we have 10 times the information), a perfect model was always found. The best tree was the same as the 1-se tree, and was based on 9 informative splits.

Further research needs to be done with this example, including

- other values of the shrinkage parameter k
- the effect of increasing observation time per subject, versus increasing the number of subjects.
- shrinking trees, as in Hastie [2]
- other measures of prediction error

One other measure of prediction error was examined briefly. We know that in the multinomial classification problem the same edge effect can occur when the deviance is used as the error measure and an observed rate is near zero or one. This can be ameliorated by using the simple sums of squares error $\| p_i - \hat{p} \|^2$, where $\hat{p}$ is the predicted probability vector for a node and p_i is the observed vector for a subject (zeros with a single 1). By analogy, we might expect $(c_i/t_i - \hat{\lambda})^2$ to avoid some of the problems with skewness associated with the Poisson deviance measure. Sadly, this did not hold true.

5 Relation to other work

One obvious use of this software is for survival data. The censoring indicator $\delta = 0, 1$ becomes the number of events for a subject, and the follow-up time is used as the time on test. In this case the likelihood ratio test for two Poisson subsamples is equivalent to the likelihood ratio test for two exponentials, and our splitting rule is the one proposed by Davis [4]. He also noticed the problem with nodes that have only a few events, leading to an infinite estimate of cross-validated error, and proposed an ad hoc shrinkage estimate for $\hat{\lambda}$. His final suggestion is to use the cross-validation results only as a guide to choosing the right tree.

LeBlanc and Crowley [5] also consider the case of survival data, but base their splitting rule on the *local full likelihood*. This procedure is equivalent to the following:

- Rescale the time values within the node so that the cumulative hazard is linear, i.e., replace each t_i with $H(t_i)$ where H is a piecewise linear estimate of the cumulative hazard.
- Use the usual exponential deviance statistic, but with the rescaled time values

As a practical matter, they suggest only rescaling the data once, at the first split. Thus, our procedure can mimic theirs simply by prescaling the data before calling the routine.

6 Software

A standalone program that implements this technique is available from statlib. Send the message "send rpart from general" to the fictitous user statlib@lib.stat.cmu.edu. The routine also can handle categorical data using the Gini criteria and regression problems using the between groups sum of squares.

A set of S functions for the same task should be submitted to statlib soon (some documentation is unfinished). People who wish to try out an early release can send mail to the author at therneau@mayo.edu.

References

[1] Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1983). *Classification and Regression Trees.* Wadsworth International.

[2] Chambers, J.M. and Hastie, T.J. (1992). *Statistical Models in S*, section 9.2. Wadsworth, Pacific Grove, Ca.

[3] Ciampi, A., Lawless, F., McKinney, S.M. and Singhal, K.(1988). Regression and recursive partition strategies in the analysis of medical survival data. *J Clin Epidemiol* **41**, 737-48.

[4] Davis, R.B. and Anderson, J.R. (1989). Exponenial survival trees. *Statistics in Med*, **8**, 947-61.

[5] LeBlanc, M. and Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics* **48**, 411-26.

[6] Segal, M.R. (1992). Tree structured methods for longitudinal data. *JASA* **87**, 628-31.

[7] Therneau, T. (1983). A short introduction to recursive partitioning. *Orion technical report 21*, Department of Statistics, Stanford University.

A Bayesian Approach to Additive Nonparametric Regression

Michael Smith and Robert Kohn, Australian Graduate School of Management
University of New South Wales, PO Box 1, Kensington 2033, Australia

Abstract

A regression model is estimated nonparametrically using regression splines to model nonlinear components with the dependent variable transformed using a Box-Cox transformation. The knots for each component, the regression variables and the data transformation are selected using a Bayesian approach with the computation carried out using the Gibbs sampler. This extends previous work on Bayesian variable selection which assumes that variables enter linearly. The performance of the proposed nonparametric estimator is applied to a number of examples and shown to work well in practice. By exploiting the special features of a spike and slab prior for the regression coefficients, our variable selection algorithm is much faster than previous Bayesian variable selection algorithms.

1 Introduction

We estimate a regression model semiparametrically using cubic regression splines to model nonlinear components. In this paper we confine the discussion to additive regression models but the approach extends in a straightforward way to a regression model with interactions. We conjecture that most nonlinear regressors observed in practice are well approximated by a regression spline with just a few knots, if those knots are carefully selected. If too many knots are used to estimate a nonlinear function which is observed with noise then a poor smooth with high local variance can result. Because, in general, we do not know how to optimally place the knots for each variable, we use many knots for each variable and select the important knots using Bayesian variable selection. We note that our approach selects which independent variables enter the regression and so extends previous work on variable selection in linear regression by Mitchell and Beauchamp (1988) and George and McCulloch (1993, 1994). We also allow the dependent variable to be transformed using a Box-Cox transformation taking a discrete number of values.

We show that our procedure works well on a number of simulated examples. In the one dimensional case we compare the nonparametric smooth obtained by Bayesian variable selection with that obtained by the kernel based locally linear least squares smoother, with the bandwidth parameter estimated by the direct plugin procedure developed Ruppert, Sheather and Wand (1993). This plugin estimator is among the best performing bandwidth estimators for locally linear least squares kernel regression.

Because of the large number of variables involved, the computation is carried out using the Gibbs sampler with the error variance, the regression parameters and the Box-Cox parameter integrated out. We place a slab and spike prior on the regression parameters and exploit this prior to obtain a fast Bayesian variable selection algorithm. When the number of variables selected is substantially smaller than the number available, which is almost always the case in our applications, then our approach can be substantially faster than that proposed by George and McCulloch (1994) who also integrate out the error variance and the regression parameters. A more detailed comparison of our approach with that of George and McCulloch (1993, 1994) is given in Section 7.

Our approach to nonparametric regression has a number of advantages over previous work. First, we just use a linear regression framework which is easy to understand and allows the usual linear regression diagnostics to be carried out after the model is estimated. Most optimal nonparametric regression estimators such as splines and kernel based nonparametric estimators are quite esoteric to the general user, especially when smoothing parameters need to be estimated as well. Second, our approach is very general and can handle additive models with interaction terms and can select the significant independent variables. At present, kernel based methods cannot handle additive models when reliable bandwidth estimation is also required. There do not seem to be at present reliable ways of doing variable selection using spline smoothing with the exception of some ad-hoc methods such as the Bruto algorithm proposed in Hastie and Tibshirani (1990, p. 262). Friedman and Silverman (1989) and Friedman (1991) also use regression splines for nonparametric regression and select the knots by a cross-validation procedure. This is computationally very intensive and makes it difficult to traverse all possible knot combinations when seeking optimal knot allocation. Hastie (1989) notes that the knot selection procedure in Friedman and Silverman (1989) can produce unsatisfactory model fits. A third advantage of our procedure is that it is very fast compared to many other nonparametric regression estimators. Except for an initial $O(n)$ calculation, our procedure is independent of sample size. Spline smoothing using either generalised cross-validation or marginal likelihood to estimate the smoothing parameter generally requires $O(n^3)$ operations, e.g. Gu and Wahba (1991) with some savings available for specialised models. Kernel based nonparametric regression requires $O(n^2)$ operations but can be considerably speeded up by using binning as in Fan and Marron (1994). Finally, our approach allows the dependent variable to be transformed as an integral part of the estimation. This can only be done on an ad-hoc basis using spline or kernel fitting.

The paper is structured as follows. Section 2 describes variable selection for linear regression and explains how the Gibbs sampler is used to find the model with the highest posterior probability. Section 3 presents our approach to nonparametric regression in the univariate case and empirically compares its performance to kernel based locally linear least squares smoothing. Section 4 generalises the treatment in Section 2 to include transformation of the dependent variable as part of the Bayesian analysis. Section 5 deals with semiparametric additive regression. Section 6 gives implementation details for variable selection and transformation of the dependent variable in a linear regression model. Section 7 compares our approach to variable selection with that of George and McCulloch (1993, 1994).

2 Variable selection in a linear regression model

In this section we review variable selection in the linear regression model as it is the basis of our nonparametric procedure. We consider the linear regression model

$$y = X\beta + e \qquad (2.1)$$

where y is the $n \times 1$ vector of observations, X is the $n \times r$ design matrix, $e \sim N(0, \sigma^2 I_n)$ is the error vector and $\beta = (\beta_1, \ldots, \beta_r)'$ is the $r \times 1$ vector of regression coefficients. Let γ be the $r \times 1$ vector of indicator variables with ith element γ_i such that $\gamma_i = 0$ means that $\beta_i = 0$ and $\gamma_i = 1$ means that $\beta_i \neq 0$. Given γ, let β_γ consist of all the nonzero elements of β and let X_γ be the columns of X corresponding to those elements of γ that are equal to one. Given γ and σ^2, we take the prior for β_γ as $\beta_\gamma|\gamma, \sigma^2 \sim N\left(0, c\sigma^2(X_\gamma' X_\gamma)^{-1}\right)$, where c is a positive scale factor specified by the user. In the empirical work we take $c = 100$ and find it performs well and makes the prior $\beta_\gamma|\gamma, \sigma^2$ almost diffuse. We take the prior of σ^2 given γ as $p(\sigma^2|\gamma) \propto 1/\sigma^2$. Finally, we take the γ_i as apriori independent with $p(\gamma_i = 1) = \pi_i$, $0 \leq \pi_i \leq 1$, for $i = 1, \ldots, r$. In our applications we take the $\pi_i = \frac{1}{2}$ which means that each model γ has a prior probability equal to 2^{-r}. Taking the π_i smaller than $\frac{1}{2}$ will result in a more parsimonious model. Our aim in this paper is to select the model with the highest posterior probability, that is the highest value of $p(\gamma|y)$. This is equivalent to maximising $p(y|\gamma)p(\gamma)$. By integrating β_γ and σ^2 out we obtain that

$$p(y|\gamma) \propto (1+c)^{\frac{1}{2}q_\gamma} S(\gamma)^{-\frac{1}{2}n} \qquad (2.2)$$

where $q_\gamma = \sum_{i=1}^r \gamma_i$ is the number of nonzero elements of β and

$$S(\gamma) = y'y - \frac{c}{1+c} y'X_\gamma (X'_\gamma X_\gamma)^{-1} X'_\gamma y \quad (2.3)$$

so that

$$p(\gamma|y) \propto (1+c)^{\frac{1}{2}q_\gamma} S(\gamma)^{-\frac{1}{2}n} \prod_{i=1}^r \pi_i^{\gamma_i}(1-\pi_i)^{1-\gamma_i}$$

To obtain the model with the highest posterior probability it is necessary to search over 2^r models. This can be done directly if r is small. In our applications r will usually be large so a direct search is not feasible and we use the Gibbs sampler (Gelfand and Smith, 1990) to traverse the parameter space. Our use of the Gibbs sampler can be described as follows.

Gibbs sampler (i) Choose an initial value $\gamma^{[0]} = (\gamma_1^{[0]}, \ldots, \gamma_r^{[0]})$ of γ perhaps by generating it from some distribution. (ii) Successively generate from $p(\gamma_i|y, \gamma_{j \neq i})$. Step (ii) is carried out many times and in two stages. The first stage is a warmup period at the end of which it is assumed that the sampler has converged to the joint distribution of $p(\gamma|y)$. The second stage is a sampling period and the γ_i collected during this period are used for inference.

We note that as the γ_i are generated, the posterior probability $p(\gamma|y)$ is also calculated (up to a constant independent of γ) so that of the models generated thus far the one with the highest posterior probability can be recorded.

The Gibbs sampler can be executed very efficiently because usually q_γ will be much smaller than r in our problems. Implementation details are given in Section 6.

3 Univariate nonparametric regression

Suppose that

$$y_i = f(x_i) + e_i \quad i = 1, \ldots, n \qquad (3.1)$$

where y_i is the th observation, e_i is an independent $N(0, \sigma^2)$ error sequence and $f(x)$ is a smooth function. We propose to approximate $f(x)$ by the cubic regression spline

$$b_0 + b_1 x + b_2 x^2 + b_3 x^3 + \sum_{k=1}^m \beta_j (x - \tilde{x}_k)_+^3, \quad (3.2)$$

where $\tilde{x}_1, \ldots, \tilde{x}_m$ are the m 'knots' placed along the domain of the independent variable x, such that $\min(x_i) < \tilde{x}_1 < \ldots < \tilde{x}_m < \max(x_i)$, while $(z)_+ = \max(0, z)$. By replacing $f(x)$ in (3.1) by its approximation (3.2) the nonparametric regression can be rewritten as a linear regression. Let $r = m + 4$, $\beta = (b_0, b_1, b_2, b_3, \beta_1, \ldots, \beta_m)'$, $\mathbf{x} = (x_1, \ldots, x_n)'$ and let $\mathbf{1}$ be a vector of n 1's. Also, let the $n \times r$ matrix $X = (\mathbf{1}, \mathbf{x}, \mathbf{x}^2, \mathbf{x}^3, (\mathbf{x} - \mathbf{1}\tilde{x}_1)_+^3, \ldots, (\mathbf{x} - \mathbf{1}\tilde{x}_m)_+^3)$. Then, with $f(x)$ replaced by (3.2), we can write (3.1) as (2.1)

The most important question associated with fitting regression splines is the choice of both the number and location of the knots $\tilde{x}_1, \ldots, \tilde{x}_m$; see, for example, Friedman and Silverman (1989) and Friedman (1991). If the knots are badly located, details of the curve can be missed, while if too many knots are included the fitted spline based on these knots will have high local variance. One way solve the problem is to introduce a large number of potential knots from which a significant subset can be selected, e.g. Friedman and Silverman (1989, pp. 9-11). The problem then becomes one of variable selection where each knot corresponds to a column of a design matrix from which a significant subset is to be determined. Although the number of knots selected, m, will typically be large so that r will be large, the number of significant variables q required to obtain a good approximation will usually be quite small. This is what makes our algorithm so fast.

We look at the performance of our approach and compare it to local linear smoothing for data sets generated from the following three curves.

$$y_i = 2x_i + e_i \qquad (3.3)$$

where $e_i \sim \text{iid}N(0, 0.5^2)$,

$$y_i = \sin(8\pi x_i) + e_i \qquad (3.4)$$

where $e_i \sim$ iid$N(0, 0.5^2)$, and

$$y_i = g(x_i) + e_i \tag{3.5}$$

where $g(x) = 10e^{-10x_i} + 2 + e_i$ if $x_i < \frac{1}{2}$ and $g(x) = 3\cos(10\pi x_i) + e_i$ if if $x_i \geq \frac{1}{2}$. In (3.5) the errors $e_i \sim$ iid$N(0, 2^2)$. One hundred observations were drawn from a Uniform(0,1) distribution, forming the independent variable for each of the three functions. The errors were also randomly generated, while the knots were chosen to follow the density of the independent variable, one every three observations. This produced a total of $m = 33$ knots and $r = 37$ columns in X from which to select. The Gibbs sampler was run for a warmup period of 300 iterations and a sampling period of 3000 iterations, with arbitrary initial condition $\gamma^{[0]} = (1, 0, \ldots, 1, 0, 1)'$. Convergence seems to have occurred within a dozen iterations for each of the three functions. When the variables selected by the Bayesian approach were placed in a linear least squares routine they were all significant at the 1% level. Figures 1(a)-1(c) show plots of the least squares fits, based on the obtained model estimates, against each set of generated data and respective true curve. Figures 2(d)-(f) show the corresponding fits obtained to the same data sets using local linear kernel based regression. Smith and Kohn (1994) repeat the above simulation 100 times and show that the three data sets generated are typical data sets for the models (3.3)-(3.5). The six plots in Figure 1 show that the regression spline estimator performs well and is smoother than the local linear estimator. This has also been our experience with other data sets.

A more extensive set of simulations and comparisons with locally linear least squares is given by Smith and Kohn (1994).

4 Data transformation

We now generalise the model (2.1) by allowing the dependent variable to be transformed using a Box-Cox transformation. Given the indicator vector γ, the linear model becomes

$$y_\lambda = X_\gamma \beta_\gamma + e \tag{4.1}$$

where $y_{i,\lambda} = y_i^\lambda$ if $\lambda \neq 0$ and $y_{i,\lambda} = \log(y_i)$ if $\lambda = 0$. As is normal when using the Box-Cox transformation, we assume that the dependent variable y_i is positive. Otherwise, some positive number is added to all the observations to make this so. In order to carry out both variable selection and transformation selection using the Gibbs sampler it will be necessary to integrate out λ. To facilitate this we allow λ to take on just a small set of values denoted by Λ. In our examples we take $\Lambda = \{-2, -1, -\frac{1}{2}, 0, \frac{1}{2}, 1, 2\}$, which will be adequate for most applications. Our aim is to find the values of λ and γ that give the highest posterior probability $p(\lambda, \gamma | y)$. To find this combination of λ and γ we run the Gibbs sampler as in Section 2 by generating from $p(\gamma_i | y, \gamma_{j \neq i})$, $i = 1, \ldots, r$. To evaluate $p(\gamma | y)$ we note that

$$p(\gamma | y) = \sum_{\lambda \in \Lambda} p(\lambda, \gamma | y) \propto \sum_{\lambda \in \Lambda} p(y | \lambda, \gamma) p(\lambda) p(\gamma)$$

and $p(y | \lambda, \gamma) = p(y_\lambda | \lambda, \gamma) J(\lambda)$, where $J(\lambda)$ is the Jacobian of the transformation $y \rightarrow y_\lambda$ and is equal to $\prod_{i=1}^n |\lambda| y_i^{\lambda - 1}$ if $\lambda \neq 0$ and $\prod_{i=1}^n \frac{1}{y_i}$ if $\lambda = 0$. From (2.2) and (2.3) we obtain

$$p(y | \lambda, \gamma) \propto (1 + c)^{\frac{1}{2} q_\gamma} S(\lambda, \gamma)^{-\frac{1}{2} n} J(\lambda) \tag{4.2}$$

where

$$S(\lambda, \gamma) = y_\lambda' y_\lambda - \frac{c}{1 + c} y_\lambda' X_\gamma \left(X_\gamma' X_\gamma \right)^{-1} X_\gamma' y_\lambda . \tag{4.3}$$

The prior for γ is the same as in Section 2 and in our applications we take a uniform prior on $\lambda \in \Lambda$.

We found it necessary to integrate λ out when generating γ. The Gibbs sampler generating $\gamma_i | y, \gamma_{j \neq i}, \lambda$, $i = 1, \ldots, r$ and $\lambda | y, \gamma$ tended to get stuck, because of the high correlation between the λ and γ iterates. If λ takes on only a small number of values then the variable selection algorithm can be very fast as the terms $y_\lambda' y_\lambda$ and $y_\lambda' X$ can all be precalculated. For each of the models generated by the Gibbs sampler it is straightforward to calculate the density $p(\lambda, \gamma | y) \propto p(y_\lambda | \gamma, \lambda) J(\lambda) p(\gamma) p(\lambda)$, up to a constant independent of λ and γ, which enables us to keep track of the values of λ and γ maximising the posterior density.

To illustrate the performance of our approach to simultaneously determining γ and λ we generated 100 observations from (3.3)-(3.5) as in Section 3. For the data generated from (3.3) we transformed $y_i \rightarrow (y_i+1)^{-\frac{1}{2}}$, for the data generated from (3.4) we transformed $y_i \rightarrow \exp(y_i + 2.5)$ and for the data generated from (3.5) we transformed $y_i \rightarrow (y_i+7)^{-2}$. Figure 5 plots the transformed data in the left panels and the original data, together with the curve estimate and the true curve, in the right hand panels for each of the three functions. It is clear, that at least for these realisations, the nonparametric approach with variable selection performs very well. Further simulations indicated that this combined approach is highly effective.

5 Additive semiparametric regression

Because regression splines are linear models it is possible to employ them in an additive model context by constructing a single design matrix made up of columns of the individual design matrices of the type outlined in the previous section. Model selection can then be performed simultaneously on the knots (and other polynomial terms) associated with each independent variable modelled by a regression spline, by selecting from the columns of this new design matrix.

The next example illustrates the performance of our approach to variable selection and data transformation on a four component additive regression model. Two hundred observations were generated from

$$\begin{aligned} y_i &= \exp\left(f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + \right. \\ &\qquad \left. f_4(x_{4i}) + e_i\right). \end{aligned}$$

The errors e_i are independent $N(0, 0.5^2)$, $f_1(z) = \sin(2\pi z), f_2(z) = -1.5z, f_3(z) = cos(6\pi z)$ and f_4 is null. The independent variables $x_{1i}, \ldots, x_{4i}$, $i = 1, \ldots, n$, are each generated from a uniform distribution. Figures 6(a)–6(d) plot y_i against each of the independent regressors and show that it is difficult to determine the functional forms $f_1, \ldots, f_4$ from these plots. The additive model

$$y_{i,\lambda} = f_1(x_{1i}) + f_2(x_{2i}) + f_3(x_{3i}) + f_4(x_{4i}) + e_i$$

was fitted to the data using the Bayesian approach explained above, with λ taking the 7 values given in Section 4. Each function f_j was approximated by a regression spline with 13 knots, one every 15 observations. We ran the Gibbs sampler with the initial value of $\gamma = (1, 0, 1 \ldots, , 0, 1)$, a warmup period of 300 iterations and a sampling period of 3000 iterations. The posterior mode of λ and γ produced a log transformation, the estimate of f_1 included linear and quadratic terms together with two extra knots, the estimate of f_2 was linear, the estimate of f_3 required the squared and cubic terms plus six extra knots and the estimate of f_4 was null. This means that out of $r = 65$ potential regressors, $\hat{q} = 14$ were selected. The R^2 for this model was 0.867. Figure 6(e) plots the transformed data (scatter plot), the true value of f_1 (solid line) and its estimate (dashed line) against x_{1i}. Figures 6(f), 6(g) and 6(h) are similar plots for f_2 to f_4, with f_4 null. These plots show that for this simulated data set our approach selects the correct data transformation and provides good estimates of the components. In particular, the null component f_4 is omitted from the model.

6 Implementing the Gibbs sampler

We outline how to efficiently implement the Gibbs sampler described in Section 2 and extend the result to the data transformation case discussed in Section 4. Before running the sampler the terms $y'y, X'y$ and $X'X$ are computed. To generate γ_i, we note that $p(\gamma_i|y, \gamma_{j \neq i})$ is binomial with $p(\gamma_i = 1|y, \gamma_{j\neq i}) = 1/(1+h)$, where

$$h = \frac{1-\pi_i}{\pi_i}(c+1)^{\frac{1}{2}}\frac{S(\gamma^1)}{S(\gamma^0)},$$

$\gamma^1 = (\gamma_1, \ldots, \gamma_{i-1}, \gamma_i = 1, \gamma_{i+1}, \ldots, \gamma_r)$ and $\gamma^0 = (\gamma_1, \ldots, \gamma_{i-1}, \gamma_i = 0, \gamma_{i+1}, \ldots, \gamma_r)$. Suppose that $\gamma = \gamma^0$ before γ_i is generated.

Then $S(\gamma^0)$ is known and it is necessary to obtain $S(\gamma^1)$. The main computational difficulty in obtaining $S(\gamma^1)$ is evaluating $y'X_{\gamma^1}\left(X'_{\gamma^1}X_{\gamma^1}\right)^{-1}X'_{\gamma^1}y$. This is done by factoring $X'_{\gamma^1}X_{\gamma^1}$ as $L_1L'_1$, where L_1 is lower triangular, using the Cholesky decomposition and then computing $L_1^{-1}X'_{\gamma^1}y$. We note that $X'_{\gamma^1}X_{\gamma^1}$ and $X'_{\gamma^0}X_{\gamma^0}$ differ by only one row and column so that L_1 can be readily obtained from L_0, where $L_0L'_0$ is the Cholesky decomposition of $X'_{\gamma^0}X_{\gamma^0}$; see Dongarra, Moler, Bunch and Stewart (1979, Ch. 10). If $\gamma = \gamma^1$ before γ_i is generated, then L_0 can similarly be obtained from L_1. From Dongarra et al. (1979), generating γ_i requires $q^2_{\gamma^1}$ operations. Hence generating γ requires $O(rq^2)$ operations, where q is the typical number of regressors required. We refer the reader to Dongarra et al. (1979) for a discussion of fast and stable methods for updating a Cholesky decomposition.

When the dependent is transformed as well, we first obtain the terms $y'_\lambda y_\lambda, X'y_\lambda$ and $X'X$ for each value of $\lambda \in \Lambda$. Fast calculation of $S(\lambda, \gamma)$ is done as above.

7 Discussion of related work

Differences in approaches to Bayesian model selection revolve primarily around the specification of the conditional prior $\beta|\gamma, \sigma^2$ because it introduces the indicator variables into the model. Mitchell and Beauchamp (1988, p.1024) use a uniform prior, letting $\beta|\gamma, \sigma^2 \sim \text{Uniform}(-a_i, a_i)$, with a_i large for each i. The decision of how large to choose the values of a_i is left to the user.

George and McCulloch (1993) use the non-conjugate normal prior $\beta_i|\gamma, \sigma^2 \sim N(0, \tau_i^2)$ if $\gamma_i = 0$ and $\beta_i|\gamma, \sigma^2 \sim N(0, c_i^2\tau_i^2)$ if $\gamma_i = 1$. The constants τ_i and c_i are chosen so that τ_i is small and c_i is large. George and McCulloch (1993) make some suggestions on suitable choices for c_i and τ_i and use the following Gibbs sampler to generate models of high probability: Generate from (a) $p(\beta|y, \sigma^2, \gamma)$; (b) $p(\sigma^2|y, \beta, \gamma)$; (c) $p(\gamma_i|y, \beta, \sigma^2, \gamma_{j\neq i})$ for $i = 1, \ldots, n$ We have found this sampler difficult to implement for our problems because of the high correlation between β and γ. If τ_i is chosen too small then the sampler is nearly degenerate and tends to get stuck. If τ_i is chosen too large, significant terms are omitted and high local bias is experienced. We note that this sampler requires $O(r^3)$ operations to generate β which can be considerably slower than our algorithm if q is much smaller than p.

George and McCulloch (1994) consider the conjugate prior $\beta_i|\gamma, \sigma^2 \sim N(0, \sigma^2\tau_i^2)$ if $\gamma_i = 0$ and $\beta_i|\gamma, \sigma^2 \sim N(0, \sigma^2c_i^2\tau_i^2)$ if $\gamma_i = 1$ and obtain $p(\gamma|y)$ by integrating out β and σ^2. Given c_i and τ_i they use the Gibbs sampler in Section 2 to generate the γ_i. The computations required are carried out efficiently using the fast Cholesky updates in Dongarra et al. (1979). Because all variables remain in the regression for each value of γ, the fast Cholesky implementation in George and McCulloch (1994) requires $O(r^3)$ operations to generate γ, which can be substantially slower than our approach which requires $O(rq^2)$ operations.

Acknowledgment

We thank Chris carter for useful discussions and Matt Wand for making his locally linear kernel regression program available to us.

References

Breiman, L., and J.H. Friedman, 1985, Estimating Optimal Transformations for Multiple regression and Correlation. *Journal of the American Statistical Association* 80, 580-598

Dongarra, J. J., C.B. Moler, J.R. Bunch and G.W. Stewart, 1979, *Linpack Users' Guide,* Philadelphia: Siam.

Fan, J. , 1992, Design adaptive nonparametric regression, *Journal of the American Statistical Association* 87, 998-1004.

Fan, J. and J.S. Marron, 1994, Fast implementation of nonparametric curve estimators. *Journal of Computational and Graphical Statistics* 3, 35-56.

Friedman J. H. , 1991, Multivariate adaptive regression splines, *Annals of Statistics* 19, 1-141.

Friedman J. H. and B.W. Silverman, 1989, Flexible parsimonious smoothing and additive modeling. *Technometrics* 31, 3-39.

Gelfand, A. E., and Smith, A. F. M., 1990, Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85, 398-409.

George, E. I. and R.E. McCulloch, 1993, Variable selection via Gibbs sampling. *Journal of the American Statistical Association* 88, 881-889

George, E. I. and R.E. McCulloch, 1994, it Fast Bayes Variable Selection. Preprint.

Gu, C. and G. Wahba, 1991, Minimising GCV/GML scores with multiple smoothing parameters vi the Newton method. *SIAM Journal of Scientific and Statistical Computing* 12, 383-398.

Hastie, T., 1989, Discussion of Flexible parsimonious smoothing and additive modeling by Friedman, J.H. and Silverman, B.W. *Technometrics* 31, 23-29.

Hastie, T.J. and R.J. Tibshirani, 1990, *Generalized additive models.* New York: Chapman Hall.

Mitchell, T. J. and J.J. Beauchamp, 1988, Bayesian variable selection in linear regression. *Journal of the American Statistical Association* 83, 1023-1036

Ruppert D., Sheather S.J., and Wand M. P., 1993, An effective bandwidth selector for local least squares regression. To appear in the *Journal of the American Statistical Association.*

Smith, M. and R. Kohn (1994) Nonparametric regression using Bayesian variable selection. Submitted for publication.

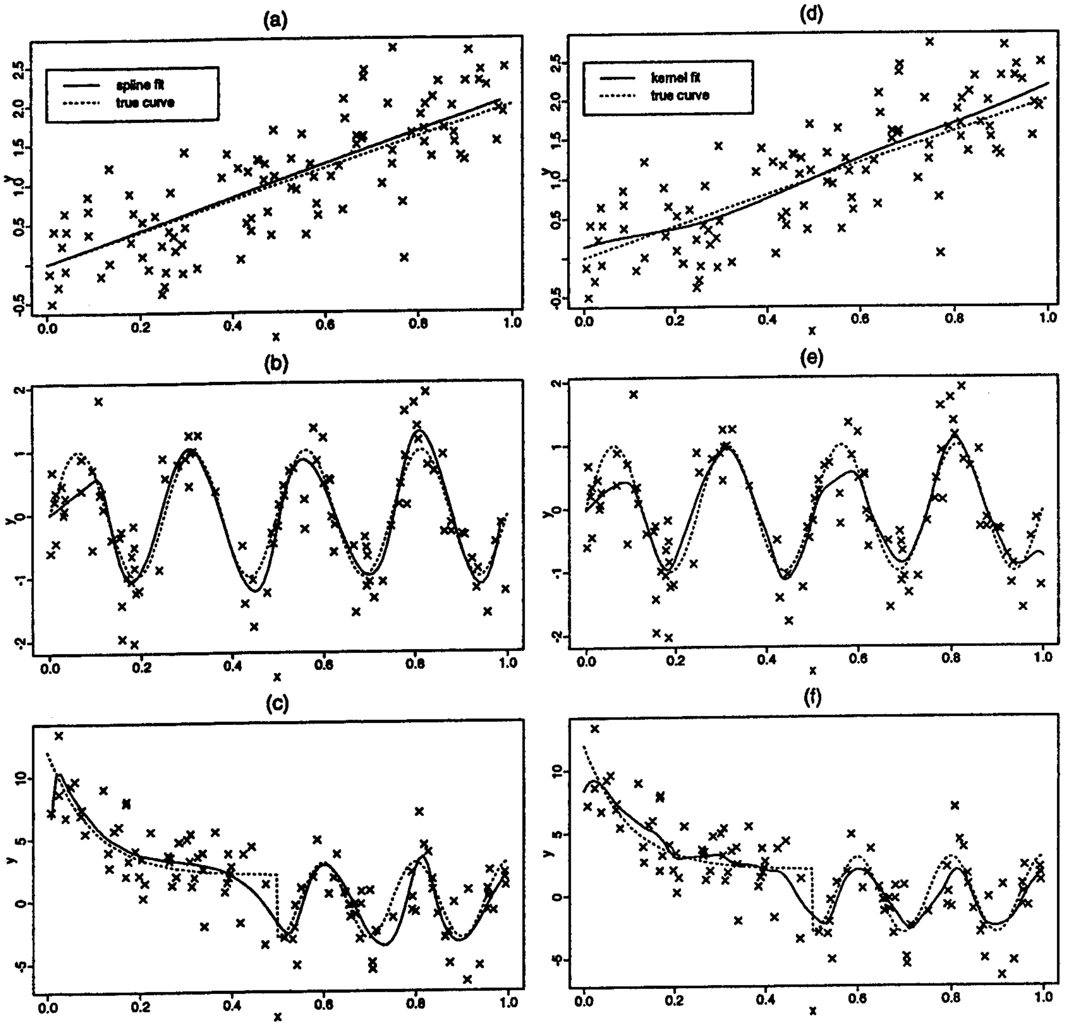


Figure 1: (a)-(c) Fits of regression splines (bold) based on model estimates for each of the three data sets (scatter plots.) The true curves (dashed) are also plotted. (d)-(f) Local linear kernel estimates (bold), the true curves (dashed) and data (scatter plots.)

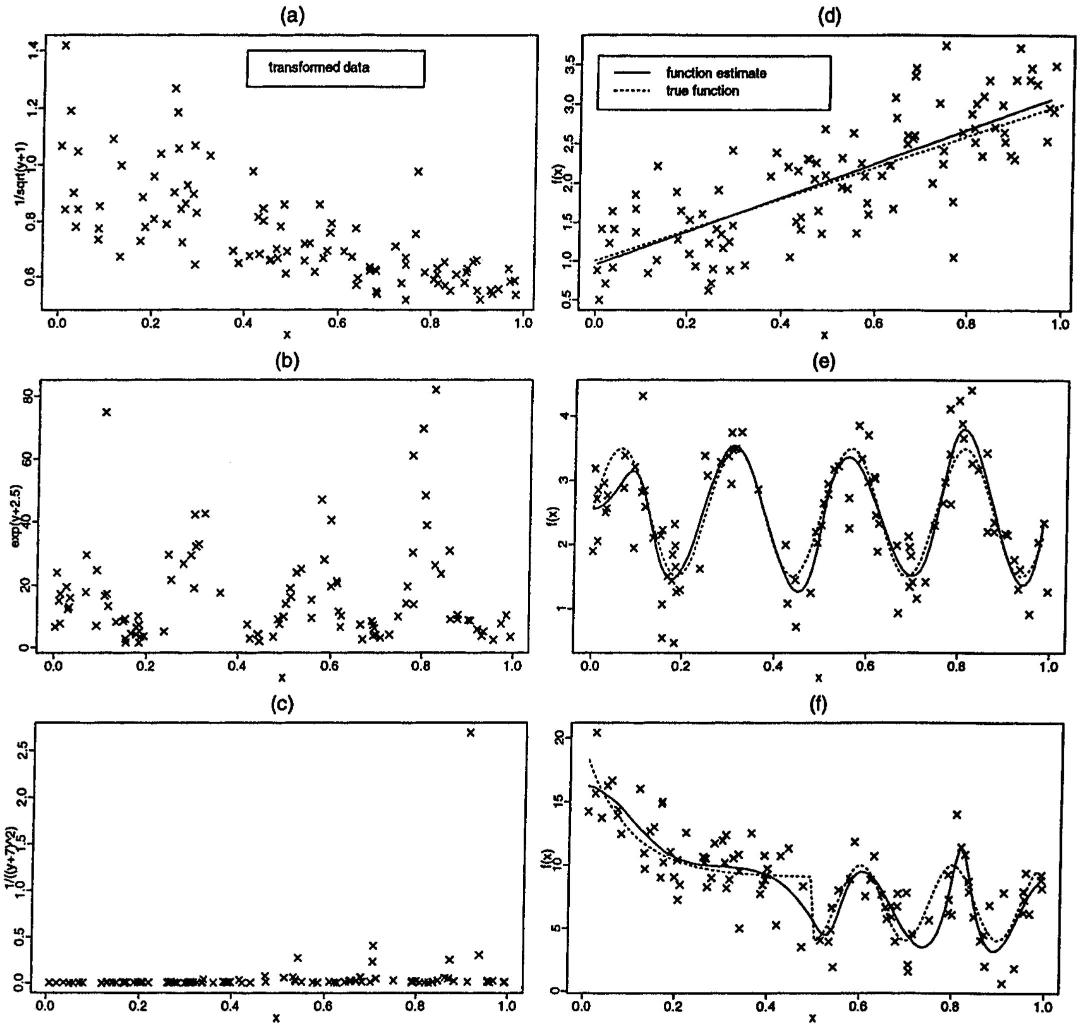


Figure 2: (a)–(c) Plots of transformed data; (d)–(f) plots of original data (scatter plot), true curves (solid) and estimated curves (dotted).

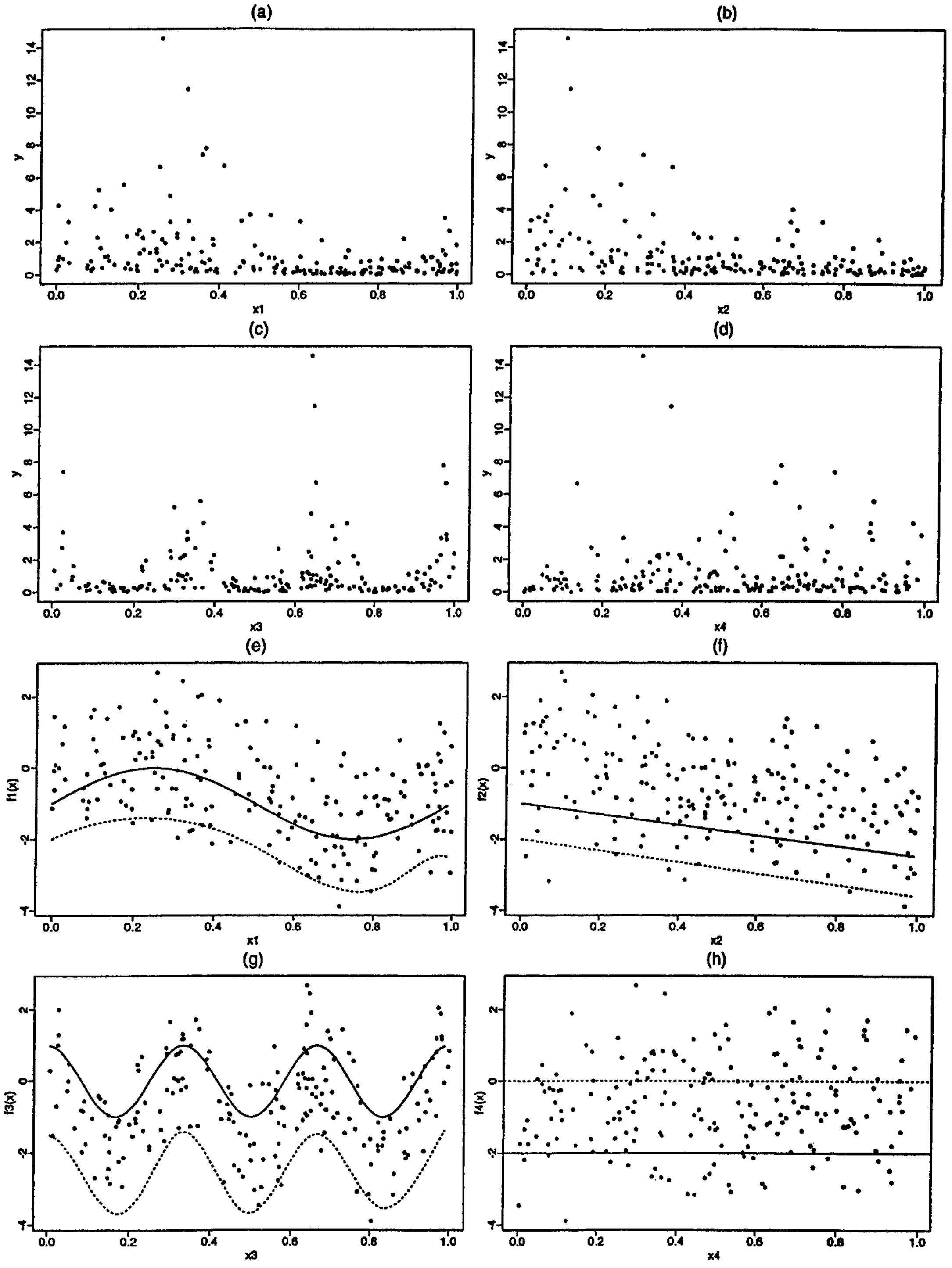


Figure 3: Parts (a)–(d) plot the transformed data against the four independent variables. Part (e) plot the transformed data (scatter plot), the true f_1 and its estimate against x_{1i}. Parts (f)-(h) are similar plots for f_2 to f_4.

Space Filling Experimental Design for Determining Protein Construct Storage Conditions

J. Alan Menius, Jr., Warren Rocque, Michael R. Emptage and S. Stanley Young
Glaxo, Inc. 5 Moore Drive, Research Triangle Park, NC 27709, U.S.A.

Abstract

Space filling experimental designs evenly distribute design points throughout a design space. These designs are useful for applications where optimums are thought to exist in distinct areas. A space filling design was carried out to determine a best set of storage conditions for a particular protein construct. The design consisted of 96 points and tested the effect of eight experimental variables on protein activity. The analysis of the results was performed using linear regression, recursive modeling, and picking the set of conditions from the experimental results which produced the highest result. None of the analysis methods were found to be completely satisfactory for the analysis of these data. This experiment, while operationally successful, demonstrates the need for better algorithms and analysis methods for generating and assessing space filling experimental designs.

Introduction

Experimental designs are used in industrial applications to determine optimal process conditions by varying many factors simultaneously. It is difficult to apply classical experimental designs when the experimental space is irregular in shape, certain experimental combinations are not physically possible, or where many of the experimental factors are categorical. Computer generated exact D-optimal designs are often used in these situations (Snee, 1985). D-optimal designs are based upon a model of the process, usually linear,

$$Y = X \beta + \varepsilon,$$

where Y is a column vector of responses, X is a design matrix , β is a column vector of coefficients to be estimated and ε is a column vector of errors from the linear model. Exact D-optimal algorithms are computer intensive; they start with a random design and then delete points from the current design and add specific points from the experimental space to maximize the determinate of the X'X matrix. The selected points tend to be on the extremes of the experimental space and it is assumed that the linear model can be used to interpolate conditions in the space.

In some situations, the experimenter is not willing to assume a model beyond saying that points near one another are going to respond similarly. In such a situation it is natural to place points throughout the space, hence the term space filling designs. Space filling designs have no underlying model and try to best fill the n-dimensional space with a finite number of points. Some algorithms for space filling experimental designs are becoming available, Kennard and Stone (1969) and SAS/QC (1993), however little attention has been given to the subsequent analysis for these designs. If there are local regions of high activity, then use of linear models is likely to be unsatisfactory. In many industrial settings, experimenters have often found that only a few of the hypothesized important factors turn out to have much of an effect, a situation called effect sparsity. If effect sparsity holds in situations where space filling designs are used, then the analysis method should find which factors are important and what regions in these subspaces have good results. There do not seem to be standard methods for finding compact regions of similar response in a high dimensional space.

We used a space filling experimental design to determine a set of storage conditions for a purified protein construct. In this design we were concerned with near neighbor predictability, dealing with an irregular sample space and estimating the pure error inherent in the assay. Various methods of analysis were used to examine the resulting dataset, including linear regression, recursive modeling and picking the maximum value from the results. While the experiment was operationally successful (a good set of conditions was found), many problems with the construction of the design and analysis were raised.

Description of Experiment

The purification and characterization of proteins is an important process in the first stages of the drug discovery process. Biotechnology is used to identify important regions of DNA and these DNA segments are spliced into a vector for expression. The resulting protein constructs are small purified segments of protein which include the active site and have the same activity as the complete parent protein. The use of purified protein constructs is becoming instrumental for determining the functionality of biochemical processes, investigating the effects of novel pharmaceuticals and solving tertiary structures of large proteins, Cunningham and Wells (1991).

In order to stabilize the constructs and maintain biological activity, purified protein constructs are stored in a buffered solution containing other chemical additives such as detergents, reducing agents, and salts. The correct combination of chemicals that make up these solutions is usually found by performing a series of experiments where the different solution additives are varied one at a time. This method requires a large number of experiments, does not have the ability to determine how the different experimental factors interact, and does not provide an estimate of experimental variation.

The factors and their ranges thought likely to contain the optimum stability conditions for a protein construct were selected by a team of protein chemists. The final list contained a total of eight storage variables. These variables and their settings were chosen based on previous experience and recent experimentation. For continuous variables a practical experimental range was determined from which three or more settings were chosen. The spacing of the settings were selected so that the gaps between settings were small enough that a narrow optimum would not be missed. In cases where only three settings were to be tested the low and high values were set inside the extreme possible conditions. Conditions were selected so that they would not interfere with subsequent experimental processes such as protein crystallography or biological assay.

The selected conditions are given in Table 1; the total number of combinations of all variables and levels produced a candidate set of 18,144 possible experimental conditions. Next, buffer/pH combinations which were not biologically or chemically practical were excluded from the candidate set. For example the MES buffers at pH higher that 6.5, the TRIS buffers at pH higher than 8.0 and the HEPES buffers where pH was below 6.5 or higher than 8.0. The exclusion of these combinations reduced the candidate set to 9720 possible experimental conditions.

Table 1

Experimental factors and ranges considered important for the optimal storage condition of purified protein constructs

Buffers	Tris, PO_4, Mes, Hepes
pH	6-9
Protein Concentration	100-1000 ug/ml
Reducing Agents	BME, TCEP, DTT
Detergents	Tween, Ethylene glycol NP-40, Octylglucoside
Temperature	-80, -20, 4 °C
NaCl	100-1000 mM
MgCl	Yes / No
Number of Possible Experiments	18,144
Number of Experiments Performed	96

Design Generation and Experimental Results

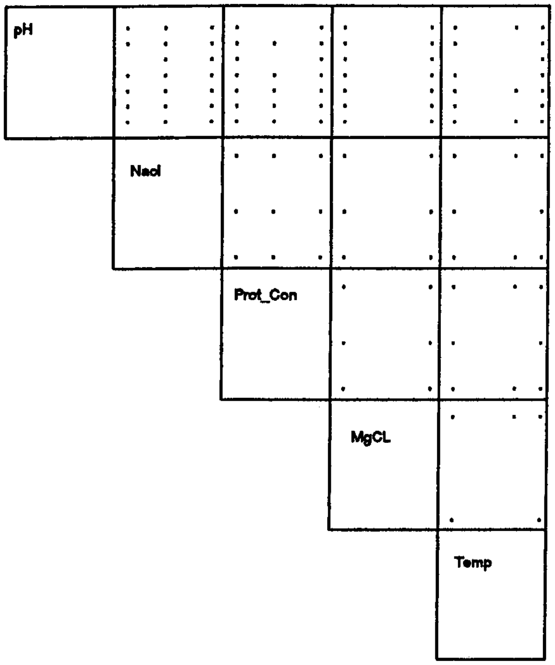
The D-optimal exchange algorithm of Mitchell and Miller (1970) as coded in Proc Optex of SAS® was used to choose design conditions from the candidate set. Main effects, quadratic effects and two way interactions were included in the model to force the algorithm to fill the experimental space. Algorithms such as those available in version 6.07 of SAS Proc Optex fill a multidimensional design space more efficiently. However at the time this experiment was performed, a satisfactory set of design points using these algorithms was unobtainable because of the large number of class variables in the design. A reference set of conditions was forced into the design and run in triplicate to give both an experimental "gold standard" and allow an estimate of pure error. The sample size for the experiment was set at 96, 93 separate conditions along with 3 replications of a "gold standard" set of conditions.

The protein activity of each of the 96 samples was determined once a week for a total of four weeks. The activity recorded after the fourth week was used for the analysis.

Design Generation Results

Two dimensional views of the numerical variables showed that in most cases there were representative points for each factor (Figure 1). However in some cases the interior areas of the design space were not well represented. Currently there are no criterion for assessing how well points fill a space.

Figure 1 Quantitative Factors



Analysis: Linear Regression

Analysis of the dataset using linear regression produced a predictive model with a large number of statistically significant predictor variables; results of this analysis are given in Table 2. Numerous two way interaction and quadratic terms were found making simple interpretation of specific experimental factors difficult. Other causes of concern were the possibility overfitting and the apparent violation of effect sparsity. Table 3 gives the predicted responses and cross validation results for the best predicted conditions. These results demonstrate that while in some cases the model was able to predict within assay error, in other cases the predicted response was erroneous, thus demonstrating the relative importance of using local points to predict in the sparse design space.

Table 2

Linear Regression Results

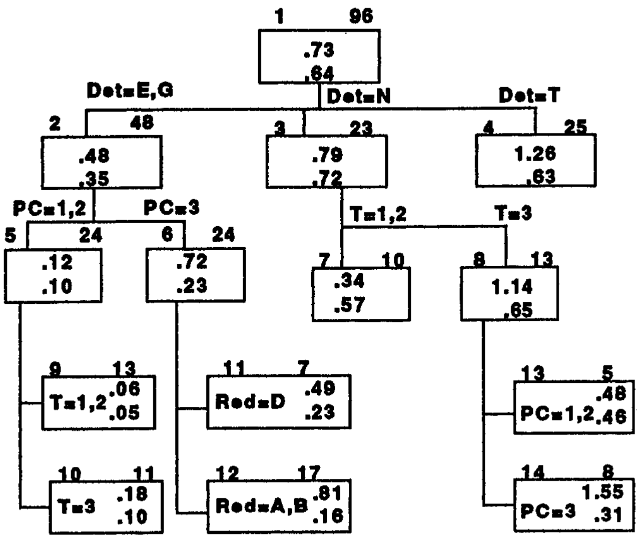
Source	df	SSq	F ratio	p Value
C. Total	95	38.46		
Buffer	3	1.47	6.12	0.0013
Buffer X NaCl	3	0.82	3.43	0.0245
Buffer X Red_Agent	6	1.45	3.01	0.0142
Buffer X Detergent	9	2.49	3.45	0.0025
Buffer X Temp	3	1.58	6.59	0.0008
NaCl	1	0.23	2.82	0.0996
NaCl X NaCl	1	0.61	7.49	0.0087
ProtConc	1	2.84	35.49	0.0001
ProtConc X ProtConc	1	0.85	10.61	0.0021
ProtConc X Detergent	3	1.65	6.86	0.0006
Red_agent	2	0.68	4.28	0.0196
Red_agent X Detergent	6	1.02	2.12	0.0689
Red_agent X Temp	2	0.82	5.10	0.0099
Detergent	3	7.85	32.64	0.0001
Detergent X Temp	3	1.93	8.03	0.0002
Temp	1	0.84	10.59	0.0021
Residual	47	3.76		

Analysis: Recursive Modeling

Recursive modeling (FIRM, Hawkins) is based on partitioning the data into two or more groups according to the range of values of one predictor. Once an initial partition is obtained, each one of the partitioned groups is divided into two or more groups based upon one the remaining predictors. The partitioning stops when the group size becomes too small to be partitioned or the group becomes homogenous. A recursive model of our data showed that the data first split with regard to which type of detergent was used. These subgroups were then split with regard to whichever variable was important (Figure 2). The optimum predicted result from the FIRM analysis had a mean predicted activity of 1.55±.31. This group of observations had higher protein concentrations, were stored at high temperatures and contained n-octoglucoside.

While recursive modeling is considered useful for finding complicated interactions in large data sets, it does have some disadvantages. FIRM creates trees by forward selection. The analysis stops when there is a non-significant split thus possibly hiding significant interactions below non-significant main effects.

Figure 2. FIRM analysis. Each box is numbered, number of observations, mean and standard deviation.



Analysis: Pick the Winner

Several of the experimental values obtained from the design points were superior to those obtained in the laboratory prior to this experiment (Table 3). The maximum result demonstrated a higher activity when compared to the gold standard value. This area of the design space should be further investigated.

Table 3

Linear Regression Model of Observed and Predicted Values of Standards and Best Experimental Results

Run	Observed	Linear Model Prediction	Cross Validation Prediction
37	2.11	2.05	1.92
76	2.06	2.03	2.01
90	1.94	1.90	1.84
20	1.88	1.55	1.01
38	1.81	1.17 *	0.89 *
65	1.79	1.83	1.91
27	1.79	1.78	1.77
79	1.73	1.70	1.67
29	1.72	1.54 *	1.44 *
68	1.69	1.35 *	0.84 *
std1	1.44	1.39	1.37
std2	1.31	1.39	1.42
std3	1.25	1.39	1.45

* = Indicates poor prediction

Discussion

Numerous methods are available for constructing space filling designs. Cluster analysis has been previously used for this purpose, Zemroch (1986). The addition of higher order terms in D-optimal methods can also be used. For example, the insertion of a quadratic term into the model will force three levels of the factor into the design. Indicator variables for categorical variables will force each category into the design. Thus a D-optimal strategy was used to construct this space filling design; now that more direct algorithms are available, they should be used. It is an open question as to which algorithms are "best" and indeed how to even measure best.

The sample size of 96 observations was chosen as the maximum amount of protein material and assay resources that were available. There was no attempt to reason how many samples were necessary to fill the sample space adequately. Such reasoning would depend upon the degree subspace considered important, effect sparsity, and the size of the gaps expected to be tolerable. Univariate gaps were considered in selecting the candidate space, but higher dimension gaps were not considered. The many categorical variables in this experiment appear to exacerbate sample size determination. The logic for selection of sample size for space filling designs remains an interesting problem.

The use of space filling experimental designs is appealing for use in industrial applications where a localized maximum or "spiked" response is expected. We were able to construct a space filling design for determining a set of storage conditions for a purified protein construct. By demonstrating superior activity to previously used conditions at several areas of the design space, we were operationally successful; however we uncovered a number of problems. At the present time the theory and software for the construction of space filling designs is not well developed. Additionally, traditional analysis procedures may not adapt well to these designs. Some of the problems that will have to be overcome include: Effect sparsity, overfitting or multiplicity, design space predictability, sample size determination and a criterion for comparing designs.

References

Cunningham, B.C. and Wells, J.A. (1991). Rational design of receptor-specific variants of human growth hormone. Proc Natl Acad. SCI. USA, 88: 3407-3411.

Hawkins, D.M. (1992). FIRM: Formal inference-based recursive modeling. University of Minnesota, School of Statistics, Technical Report 546.

Kennard, R.W., and Stone, L.A. (1969). Computer aided design of experiments. Technometrics, 11:137-148.

Mitchell, T.J. and Miller, F.L. (1970). Use of design repair to construct designs for special linear models. Math. Div. Ann. Report (ORNL-4661), 130-131, Oak Ridge National Laboratory.

SAS Institute, Inc. (1993). SAS/QC Software Reference. Cary, NC: SAS Institute Inc.

Snee, R.D. (1985). Computer-aided design of experiments - Some practical experiences. Journal of Quality Technology 17:222-236.

Zemroch, P.J. (1986). Cluster analysis as an experimental design generator, with application to gasoline blending experiments. Technometrics 28:39-49.

Analysis of Space-Filling Designs

Perry Haaland, Becton Dickinson Research Center, Research Triangle Park , NC,
Nancy McMillan, National Institute of Statistical Sciences, Research Triangle Park, NC,
Douglas Nychka, Department of Statistics, North Carolina State University, Raleigh, NC,
and William Welch, Department of Statistics and Actuarial Sciences, University of
Waterloo, Waterloo, Ontario

Abstract

In this paper we evaluate the usefulness of the following nonparametric regression methods for the analysis of a space-filling design: Gaussian stochastic process models, thin-plate splines, single hidden layer neural networks, generalized additive models and multiple adaptive regression splines. The space-filling design on which the evaluation is based was used to optimize the buffer for a DNA amplification method. The methods were evaluated based on how well they fit the data and on the reasonableness of the resulting multidimensional structures. The methods of Gaussian stochastic processes and thin-plate splines seemed most useful for this data set.

1 Introduction

Space-filling designs have been primarily used for computer experiments [1] [2]. However, we believe that these designs should also be useful in physical experiments in the pharmaceutical and biotechnology industries. For example, the use of a space-filling design has been reported by Menius and Young [3] where it was used to discover storage buffer conditions that preserved the activity of a protein construct, and Van Cleve [4] carried out the space-filling design which we analyze in this paper in order to optimize buffer conditions for a DNA amplification method.

One of the assumptions motivating the use of a space-filling design is that the response surface is likely to be highly nonlinear. Thus, a low order polynomial model, as traditionally used for a response surface design, will not be sufficiently flexible to capture the relevant structure of the underlying surface. Consequently, flexibility in the regression model is critical. In this paper we explore the use of several methods which can loosely be classified as nonparametric regression surfaces because of the highly flexible nature of their regression models. These methods include Gaussian stochastic processes, thin-plate splines, single hidden layer neural nets,, generalized additive models, and multiple adaptive regression surfaces.

The goal of the analysis of a space-filling design is to fit a model

$$Y = f(\mathbf{x}) + \epsilon$$

where Y is a response variable and $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T$ is the set of experimental or predictor variables. In the context of process optimization, it is of interest to find the best settings of the subset of important variables and to predict the response value at these optimum settings. Scientists and engineers may also gain insight into the underlying mechanisms by examining the structure of the surface generated by f.

Two fundamental obstacles to this process are that the form of the function f is generally unknown and that d is usually large. Because the form of f is unknown, an approximating function of some sort must be used. Consider the case in which f is approximated by a m^{th} order polynomial; then f will have

$$\begin{pmatrix} m+d \\ m \end{pmatrix}$$

terms, growing like m^d. The exponential increase in terms as a function of the dimension is known as the curse of dimensionality and this difficulty affects all approaches to the problem.

In order to be useful for the analysis of a space-filling design, a nonparametric regression model must be flexible enough to capture the multidimensional structure of the surface. In this paper, we evaluate the nonparametric regression models with this criterion in mind. The example used for the evaluation is described in the next section. In Section 3 we provide a brief description of each of the regression models we evaluated. Section 4 presents the results of the model fitting and makes comparisons among the different methods. Finally, Section 5 provides some brief conclusions.

2 Strand Displacement Amplification

Strand displacement amplification (SDA) is a method for DNA amplification that was invented at the Becton

Dickinson Research Center [5] and [6]. SDA is an isothermal amplification method that utilizes the ability of an enzyme to nick an unmodified strand of a hemiphosphorothioate form of DNA at its recognition site. DNA polymerase extends the nicked site and displaces the downstream DNA strand. Exponential amplification results from coupling sense and antisense reactions in which strands displaced from a sense reaction serve as a target for an antisense reaction and vice versa.

A space-filling design was conducted to study the effects of buffer composition on strand displacement amplification by Van Cleve [4]. Four buffer components were systematically varied in a space-filling design; namely, KCl, KPO_4, $MgCl_2$ and dNTP. The two components KCl and KPO_4 are thought to affect amplification primarily via their contribution to the ionic strength of the buffer. Ionic strength affects DNA hybridization and enzyme activity. Each enzyme in the system is likely to have its own optimal salt concentration so there may well be several local optima. The variable dNTP (deoxyribonucleotides) represents the micromolar concentration of each of the four basic building blocks of DNA needed for extending the nicked site. dNTP binds Mg^{+2} one a one-to-one basis so $MgCl_2$ must be present in at least equal molar concentration as the total dNTP concentration in order for extension to take place. Since Mg^{+2} is also a cofactor for the restriction enzyme, it needs to be in excess of the total dNTP concentration.

The design was constructed in three stages. First, a 2000 run Latin hypercube design was generated. Second, each factor was rounded to 20 levels. These 2000 runs were the candidate set of design points. Third, the best settings from previous experiments were specified as a fixed point and 55 additional design points were selected based on an approximation to the maximin criteria of Johnson, Moore, and Ylvisaker [8]. The software ALEX (ALgorithms for Efficient eXperiments, Welch [9]) was used to generate the design. Some of the runs were replicated and a total of 89 response values on the 56 different buffers were available for analysis

The response value is the counted intensity of an appropriate band on an electrophoresis gel evaluated on a PhosphorImager (Molecular Dynamics Model 425E). Because not all of the experimental buffers could be evaluated on one gel, two replicates of a control condition were run on each gel. The control condition represented the best buffer from previous experiments. (The corresponding settings were included as the fixed point when generating the design as described above.) The values were normalized for each gel as follows:

$$y_{ij}^{normalized} = y_{ij}\frac{\bar{c}}{\bar{c}_j}$$

where $\bar{c}$ is the average of all of the control runs, and $\bar{c_j}$ is the average of the control runs from gel j. Because of the exponential amplification, it makes sense to analyze the response values on the log scale in order to get at the actual amplification rate. In this context, the normalization can be regarded as a forced additive day effect.

3 Nonparametric Regression Models

Nonparametric regression can be thought of as a general class of methods that provide very flexible approximating functions. In this section we describe the methods that were used to model the data from the experiment described in the previous section.

3.1 Polynomial Models

The polynomial regression model can be expressed as follows

$$Y = \mathbf{p}^T(\mathbf{x})\boldsymbol{\beta} + \epsilon,$$

where $\mathbf{p}$ is a vector of polynomial linear model terms, $\boldsymbol{\beta}$ are the usual linear regression parameters and ϵ is the random error. An m^{th} order polynomial will include power and cross terms up to order m. As m increases, the flexibility of the polynomial model increases at the expense of possible overfitting. The *lm* function in S-PLUS [7] was used to fit the polynomial models.

3.2 Gaussian stochastic processes

In the Gaussian stochastic process model, we model Y by

$$Y = \mathbf{p}^T(\mathbf{x})\boldsymbol{\beta} + Z(\mathbf{x}) + \epsilon,$$

where $\mathbf{p}$ is a vector of linear model terms, $\boldsymbol{\beta}$ is the vector of corresponding (unknown) coefficients, $Z(\cdot)$ is assumed to be a univariate Gaussian stochastic process on the design space, and ϵ, representing random measurement error, is assumed independent of $Z(\cdot)$ and Gaussian with mean zero. In the model fit in this work, $\mathbf{p}^T(\mathbf{x})\boldsymbol{\beta}$ is simply $1\beta_0$. In this case, systematic dependence of Y on $\mathbf{x}$ is captured solely by the $Z(\mathbf{x})$ term.

Flexible specification of $Z(\cdot)$ is key to capturing the features of complex response surfaces. We consider only mean zero Gaussian stochastic processes with correlation functions of the form,

$$\begin{aligned} R(\mathbf{x},\mathbf{x}') &= \mathrm{Cor}(Z(\mathbf{x}), Z(\mathbf{x}')) \\ &= \prod_{k=1}^{d} \exp\{-\theta_k |x_k - x_k'|^{p_k}\}. \end{aligned}$$

Previous work, Sacks, Welch, Mitchell, and Wynn [1] and Welch, Buck, Sacks, Wynn, Mitchell, and Morris [2], has found this structure to be sufficiently flexible to capture quite complicated response surfaces. McMillan, Sacks, Welch and Gao [10] also reported the successful use of this method to analyze a space-filling design. The essential idea behind this covariance structure is that points "near" each other in the design space should be more correlated than points "far" from each other with the measure of "nearness" being individually scaled in each dimension of the design space. We note that this model is a universal kriging model with the covariance structure specified via R rather than the traditional variogram. (See Cressie [11] for a review of kriging.) The correlation function, R, is more general than variograms typically found in the kriging literature (where d is only 2 or 3) as R does not assume isotropy.

Now operationally, suppose we have n observations of the system, $(Y_1, \mathbf{x}_1), \cdots, (Y_n, \mathbf{x}_n)$. Let the vector of responses, $(Y_1, \cdots, Y_n)^T$ be denoted by $\mathbf{Y}$. The model we've described for this data can be written in matrix notation as

$$\mathbf{Y} = \mathbf{P}\boldsymbol{\beta} + \mathbf{Z} + \boldsymbol{\epsilon},$$

where $\mathbf{P}$ is the expanded design matrix with $\mathbf{p}^T(\mathbf{x}_i)$ in the ith row, $\mathbf{Z} = (Z(\mathbf{x}_1), \cdots, Z(\mathbf{x}_n))^T$ is the vector of stochastic process values at the n experimental settings, and $\boldsymbol{\epsilon} = (\epsilon_1, \cdots, \epsilon_n)^T$ is the vector of random errors. We assume $\mathbf{Z} \sim \mathrm{N}(\mathbf{0}, \sigma_Z^2\mathbf{R})$, where the $n \times n$ matrix $\mathbf{R}$ has $R(\mathbf{x}_i, \mathbf{x}_j)$ as the (i,j) element, $\boldsymbol{\epsilon} \sim \mathrm{N}(\mathbf{0}, \sigma_\epsilon^2\mathbf{I})$, and $\mathbf{Z}$ and $\boldsymbol{\epsilon}$ are independent. These assumptions imply $\mathbf{Y} \sim \mathrm{N}(\mathbf{P}\boldsymbol{\beta}, \sigma^2\mathbf{C})$, where $\sigma^2 = \sigma_Z^2 + \sigma_\epsilon^2$, and the $n \times n$ correlation matrix $\mathbf{C}$ is given by $(\sigma_Z^2\mathbf{R} + \sigma_\epsilon^2\mathbf{I})/\sigma^2$.

When R and $\sigma_\epsilon^2/\sigma^2$ are assumed known, the best linear unbiased predictor (BLUP) of $Y(\mathbf{x})$ is

$$\hat{Y}(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\hat{\boldsymbol{\beta}} + \mathbf{c}(\mathbf{x})^T\mathbf{C}^{-1}(\mathbf{Y} - \mathbf{F}\hat{\boldsymbol{\beta}}).$$

Here $\mathbf{c}(\mathbf{x})$ is a vector with element i given by $\frac{\sigma_Z^2}{\sigma^2}R(\mathbf{x}, \mathbf{x}_i)$, the correlations between the Y's at $\mathbf{x}$ and the n experimental runs. The vector of coefficients, $\hat{\boldsymbol{\beta}}$, is the generalized least squares estimator, $\hat{\boldsymbol{\beta}} = (\mathbf{P}^T\mathbf{C}^{-1}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{C}^{-1}\mathbf{Y}$.

We use the estimator $\hat{Y}$ to make predictions regarding the response surface for optimization purposes. First, though, we must handle the difficulty that the parameters of the covariance structure, $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_d)$, $\mathbf{p} = (p_1, \cdots, p_d)$, σ_Z^2, and σ_ϵ^2, are not known. Available software (ALEX [9] – implemented by Welch) performs maximum likelihood estimation of these quantities. Estimates of $\boldsymbol{\theta}$, $\mathbf{p}$, σ_Z^2, and σ_ϵ^2, thus obtained, are used in $\hat{Y}$ for optimization of the predictor.

As one last issue about this model we evaluate the fit of the model to the data by an empirical measure of MSE averaged over the design points,

$$\mathrm{MSE} = \frac{1}{n - \mathrm{tr}(\mathbf{H})}\sum_{i=1}^{n}(\hat{Y}(\mathbf{x}_i) - Y(\mathbf{x}_i))^2.$$

The "hat" matrix, $\mathbf{H}$, is defined by

$$\begin{aligned} \mathbf{H} &= (\mathbf{I} - (\sigma_z^2/\sigma^2)\mathbf{R}\mathbf{C}^{-1})\mathbf{P}(\mathbf{P}^T\mathbf{C}^{-1}\mathbf{P})^{-1}\mathbf{P}^T\mathbf{C}^{-1} \\ &\quad + (\sigma_z^2/\sigma^2)\mathbf{R}\mathbf{C}^{-1}. \end{aligned}$$

We use the trace of the "hat" matrix as a surrogate for the degrees of freedom in the model as suggested by Wahba [12] for splines.

3.3 Thin-plate splines

The mth-order thin-plate splines approximating function f is the minimizer of the following quantity

$$\mathcal{L}(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \rho\mathcal{J}_{m,d}(f)$$

where $\rho > 0$, f has square integrable partial derivatives up to degree m,

$$\mathcal{J}_{m,d}(f) = \sum\nolimits_{\alpha_1+\ldots+\alpha_d=m} \left(\begin{array}{c} m \\ \alpha_1 \ldots \alpha_d \end{array} \right) \int \left\{ \frac{\partial^m}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}} f(\mathbf{x}) \right\}^2 d\mathbf{x}$$

and $\mathcal{J}_{m,d}(f) < \infty$ (Wahba [12] and Nychka, Ellner McCaffrey and Gallant [13]). $\mathcal{J}_{m,d}(f)$ is a general (rotation) invariant measure of the roughness in the function f and by varying the value of ρ we can control the smoothness of the regression surface. The value of ρ is usually chosen by cross-validation.

The solution to the thin-plate spline minimization problem will be a linear combination of $\left(\begin{array}{c} m+d-1 \\ m \end{array} \right)$ monomials up to degree $m-1$ and n radial basis functions. The coefficients in this linear combination are linear functions of $\mathbf{Y}$. Therefore, there exists an implicit smoother matrix $S(\rho)$ such that $\hat{\mathbf{Y}} = S(\rho)\mathbf{Y}$ where $S(\rho)$ depends on ρ, m, d and the $\mathbf{x}_i$'s but not on $\mathbf{Y}$. The effective degrees of freedom for the regression model can be approximated by $\operatorname{tr}(S(\rho))$ and the mean square error estimated as in a similar manner as described for the Gaussian stochastic process model (Nychka [14]). The thin-plate spline model was fit using a FORTRAN function callable from S-PLUS (*tpsreg* [15] - implemented by Nychka).

For the purposes of interpretation, the thin-plate spline model can be thought of as a limiting case of the Gaussian stochastic process model. In particular, if we take $\theta_i \equiv \theta$ and $p_i \equiv 2$ in the Gaussian stochastic process model, the thin-plate spline model arises as we take the limit as $\theta \rightarrow 0$. The order of the thin-plate spline is by default $m = (d+2)/2$ so that the polynomial part of the model $\mathbf{p}^T$ is of degree m.

3.4 Neural Networks

Single hidden layer neural networks can be viewed as nonlinear regression models; namely,

$$Y = \beta_0 + \sum_{j=1}^{m} \beta_j f(\gamma_{j0} + \sum_{k=1}^{d} \gamma_{jk} x_k) + \epsilon$$

where the function f simulates the on/off firing of a single neuron. It is important that it is sigmoidal and bounded. We take f to be the the usual squashing function (logistic distribution function) $f(u) = e^u/(1+e^u)$ and $j = 1, \ldots, m$ are units (nodes) in a single hidden layer, γ_{jk} are the input weights for each node, β_j are the weights of the hidden units, and β_0 and γ_{j0} are bias adjustments (constants). The neural network model was fit using a FORTRAN program callable from S-PLUS (*nnreg* [15] - implemented by Nychka).

The neural net seems to be a good model for many problems including nonlinear regression (Cheng and Titterington [16], Geman, Bienenstock, and Doursat [17] and Nychka, Ellner, McCaffrey, and Gallant [13]). However, the $(1 + m(d+2))$ parameters are estimated by nonlinear least squares and it is often difficult to find a global minimizer. Dimension reduction occurs because of the ability to look at linear combinations of many variables. There is similarity to the method of projection pursuit in which f is replaced by arbitrary functions.

3.5 Generalized Additive Models

An generalized additive model (GAM) [18] approximates the regression surface by a function of the following form:

$$Y = \alpha + \sum_{j=1}^{d} f_j(x_j) + \epsilon$$

where each function f is a nonparametric smoothing function. It is also possible to specify a family for the error distribution, and we used the standard gaussian family in this example. A variety of smoothing functions can be used, and we used smoothing splines with 4 degrees of freedom where the degrees of freedom is equal to $tr(S) - 1$ where S is the implicit smoother matrix. Note that it is also possible to include products of the smoothing functions in the model to fit a more complex surface. We considered models involving up to linear-by-linear interactions and quadratic functions of the smoothers to fit this example. The generalized additive models were fit using the function *gam* [19] in S-PLUS [7].

3.6 Multiple Adaptive Regression Splines

Multiple adaptive regression splines (MARS) models were proposed by Friedman [20]. The regression surface is approximated by a function of the following form:

$$Y = \beta_0 + \sum_{j=1}^{m} \beta_j \prod_{l=1}^{L_j} h_{jl}(x_{v(j,l)}) + \epsilon$$

where the h_{jl} are piecewise linear basis functions. The value of $v(j,l)$ is an index of the predictor used in the lth term of the jth product. The basis functions h_{jl} are defined in pairs:

$$\begin{aligned} h_{jl}(x) &= [x - t_{jl}]_+ \\ h_{j,l+1}(x) &= [t_{jl} - x]_+ \end{aligned}$$

for l an odd integer, where the knot value is one of the unique values of $x_{v(j,l)}$. The model is constructed in a forward stepwise manner followed by pruning of the least important terms. The *degree* of the MARS fit specifies the maximum number of terms allowed in any product and so controls the level of interactions among predictor variables. MARS takes advantage of any low order structure in the response surface and generally adds terms to the model parsimoniusly. The MARS models were fit using the *mars* function in the *fda* library implemented in S-PLUS [7] by Hastie, Tibshirani and Buja [21].

4 Model Fitting Results

Each of the nonparametric regression models described in the previous section was fit to the experimental data. In cases where there were choices regarding the degree of the model, a number of alternatives were considered; in particular, polynomial models of degree 1-4 were fit, neural nets were fit with 2-5 hidden units, generalized additive models (GAM) with were fit with linear smooths, cross-products of linear smooths and powers of linear smooths, and multiple adaptive regression spline (MARS) models of degree 1-5 were fit.

The best model fitting results for each method are given in Table 1. The polynomial, Gaussian stochastic process, thin-plate splines and single hidden layer neural network models each give a satisfactory fit to the data. However, the GAM and MARS models do not appear to be flexible enough to adequately represent the data.

Table 1: Model Fitting Results

Model	Desc.	DFr	DFe	RMSE	R^2
Polyn.	Degree=4	55	34	0.83	94%
GaSP		44	45	0.83	93%
TPS	Order=3	53	36	0.83	94%
NNet	Units=5	31	58	0.77	91%
GAM	lin. + tfi	28	61	1.41	84%
MARS	Degree=3	12	77	1.66	48%
Pure Error			33	0.84	

In order to further compare the four best models on how well they capture the multidimensional structure of the example, we ran an optimization routine (to maximize the predicted response) starting at the conditions of the best design point. The results are shown in Table 2. Note that Gaussian stochastic process model and thin-plate splines give results which are quite similar to the settings of the best run. However, the neural net and polynomial models find optimal settings that are far from the starting point and have implausible predicted response values.

Table 2: Optimization Results for Best Models

Model	KCL	$MgCl_2$	KPO_4	dNTP	yopt
best run	35	6	20	1000	13.7
GaSP	36	6.2	21	975	13.4
TPS	34	6.1	20	975	13.4
NNet(5)	17	7	21	1500	22.6
Polyn(4)	50	6.8	45	650	454

Further insight into the usefulness of the models can be gained by examining response surface and contour plots for each of the four best models. These can be seen in Figure 1-4. The Gaussian stochastic process surface (Figure 1) and the thin-plate spline surface (Figure 2) are quite similar. However the Gaussian stochastic process surface suggests that there is a local optima in addition to the global optima. The local optima is close to the location of the previous best runs (control settings) and so seems plausible. The neural net surface (Figure 3) apparently is not sufficiently flexible to represent the optimum and instead suggests a rising ridge. (This is why the predicted optimum moved away from the best run.) Finally, the polynomial model (Figure 4) introduces large variations in the surface away from the data points in order to obtain a good fit. The overall surface, consequently, is not reasonable due to the overfitting.

5 Conclusions

The analysis of space-filling designs is challenging because of the flexibility required in the approximating function. It is important to fit the observed data well while avoiding overfitting and at the same time providing a reasonable representation of the multidimensional structure of the surface. In this paper we used a number of very flexible models which we loosely classified as nonparametric regression models to fit the data from a buffer optimization example.

Due to the wide range of surfaces which might be encountered, it is unlikely that there is one best method for the analysis of space-filling designs. In this example, the methods of Gaussian stochastic processes, thin-plate splines, single hidden layer neural networks and polynomial models all provided good fits to the data. Generalized additive models and multiple adaptive regression splines did not fit the data well. A more detailed examination of the multidimensional structure of the fitted surfaces showed that only the Gaussian stochastic process and thin-plate spline models provided reasonable surfaces.

In conclusion, we feel that the use of space-filling designs provides a promising approach for a wide class of problems in the pharmaceutical and biotechnology industries in which it is necessary to model complex nonlinear surfaces. As we gain more experience with these designs and their analysis, we hope to be able to more clearly identify their strengths and weaknesses, offer guidelines for their effective use, and recommend methods for nonparametric regression methods for modeling the surface structure.

References

[1] Sacks, J., W.J. Welch, T.J. Mitchell, and H.P. Winn (1989). "Design and analysis of computer experi-

ments," *Statistical Science* **4**, 409-435.

[2] Welch, W.J. , R.J. Buck, J. Sacks, H.P. Wynn, T.J. Mitchell, M.D. Morris (1992). "Screening, predicting, and computer experiments," *Technometrics* **34**, 15-25.

[3] A. Menius, and S. Young (1994). "Design of a space-filling experiment to optimize buffer conditions for a protein construct." submitted to *Journal of Biopharmaceutical Sciences*

[4] M. Van Cleve, Becton Dickinson Research Center, personal communication.

[5] Walker, G.T., M.S. Fraiser, J.L. Schram, M.C. Little, J.G. Nadeau, D.P. Malinowski (1992). "Strand displacement amplification – an isothermal, *in vitro* DNA amplification technique." *Nucleic Acids Research* **20**, 1691-6.

[6] Spargo, C.A., P.D. Haaland, S.R. Jurgensen, D.D. Shank, G.T. Walker (1993). "Chemiluminescent detection of strand displacement amplified DNA from species comprising the Mycobacterium tuberculosis complex." *Molecular and Cellular Probes* **7**, 395-404.

[7] *S-PLUS.* The StatSci Division of MathSoft, Inc., 1700 Westlake Ave. N., Suite 500, Seattle, WA 98109.

[8] Johnson, M.E., Moore, L.M., and Ylvisaker, D. (1990). "Minimax and maximin distance designs," *Journal of Statistical Planning and Inference*, 26, 131–148.

[9] *ALgorithms for Efficient eXperiments,* William Welch, University of Waterloo, Waterloo, Ontario.

[10] McMillan, N.J., J. Sacks, W.J. Welch, F. Gao (1994). "Gaussian stochastic process models," submitted to *The Journal of Biopharmaceutical Sciences.*

[11] Cressie, N.A.C. (1991). *Statistics for Spatial Data,* John Wiley & Sons, Inc, New York.

[12] Wahba, G. (1990). *Spline models for observational data,* SIAM, Philadelphia.

[13] Nychka, D, S. Ellner, and A. R. Gallant (1992). "Finding chaos in noisy systems," *Journal of the Royal Statistical Society, B* **54** 399-426.

[14] Nychka D. (1990). "The average posterior variance of a smoothing spline and a consistent estimate of the average squared error," *Annals of Statistics* **18**, 415-428.

[15] *FUNFITS,* Douglas Nychka, Department of Statistics, North Carolina State University, Raleigh, NC.

[16] Cheng, B. and D.M. Titterington (1994). "Neural networks: A review from a statistical perspective" (with comments), *Statistical Science* **9** 2-54.

[17] Geman, S. E. Bienenstock and R. Doursat (1992). "Neural networks and the bias/variance dilemma," *Neural Computation* **4** 1-58.

[18] Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models.* Chapman and Hall, London.

[19] Chambers, J.M. and T. Hastie (editors) (1992). *Statistical Models in S,* Wadsworth, Pacific Grove, CA.

[20] Friedman, J. (1991). "Multiple adaptive regression splines (with discussion)," *Annals of Statistics* **19(1)**, 1-141.

[21] Hastie, T., R. Tibshirani, and A. Bujas (1993). "Flexible Discriminant Analysis by Optimal Scoring," AT&T Bell Laboratories Manuscript. The *fda* library of functions including the *mars* function is available from the statistics archive at Carnegie-Mellon University (**statlib@lib.stat.cmu.edu**).

Figure 1. Response Surface for GaSP Model

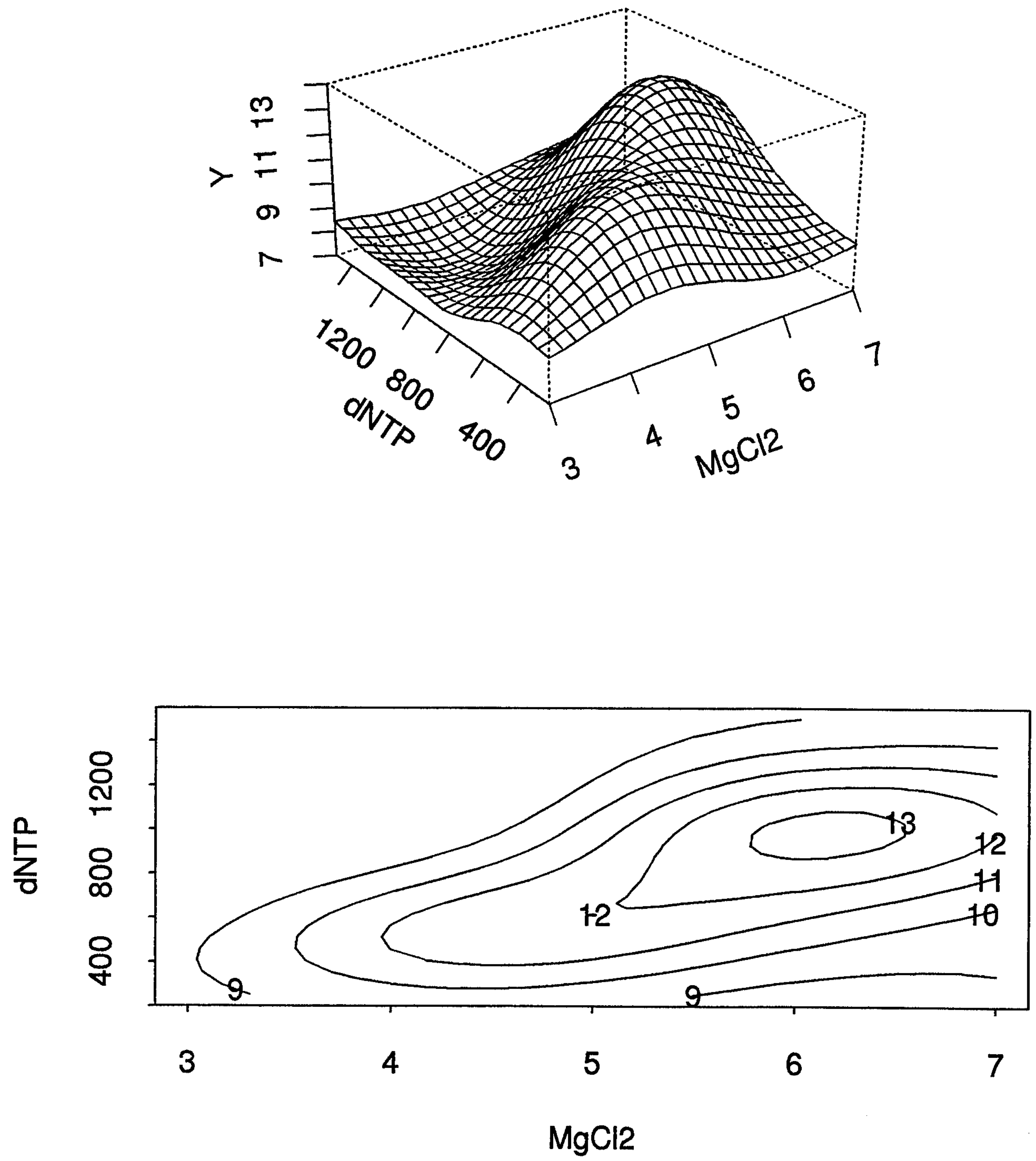


Figure 2. Response Surface for Thin-Plate Spline Model

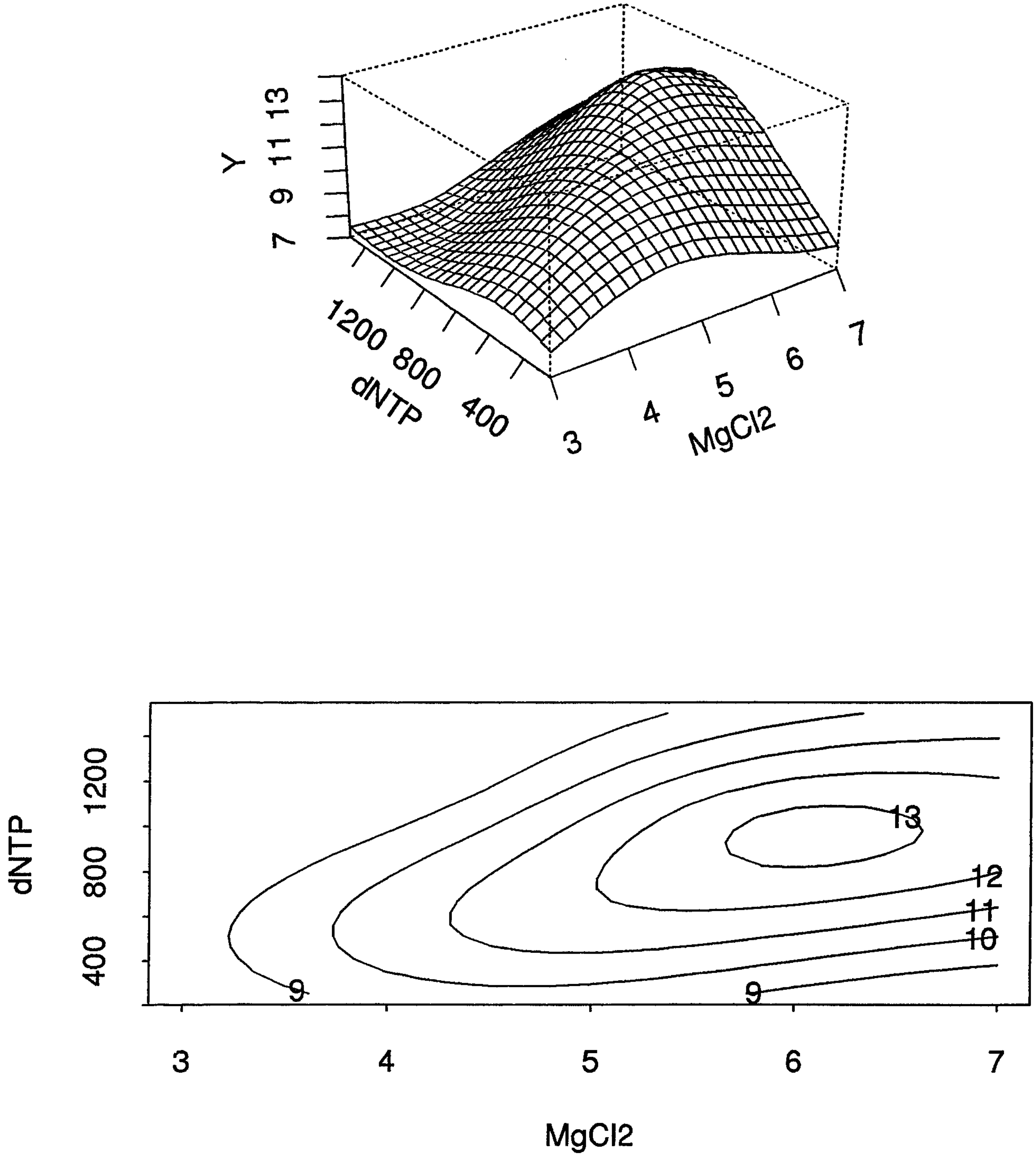


Figure 3. Response Surface for Neural Net Model

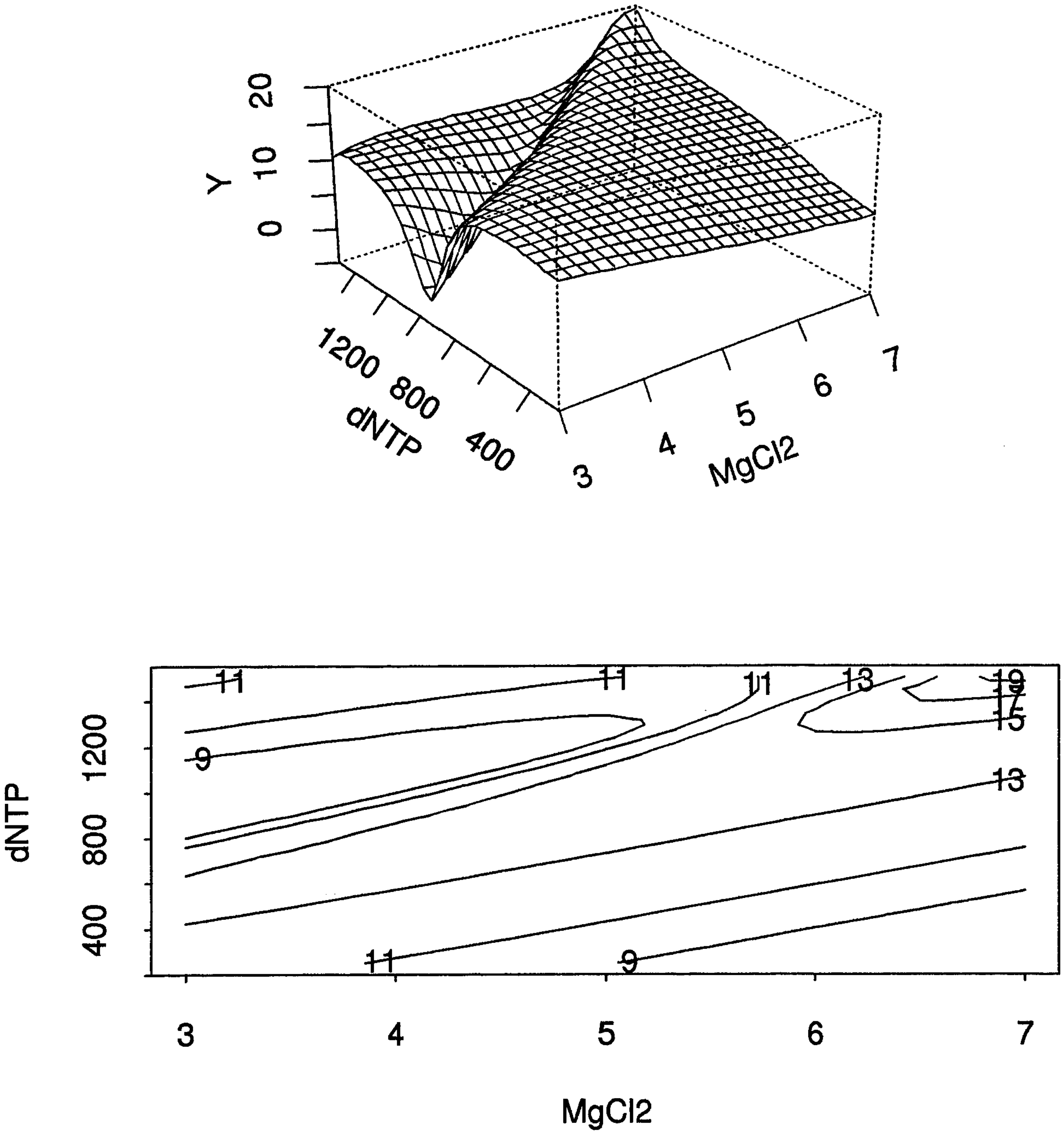
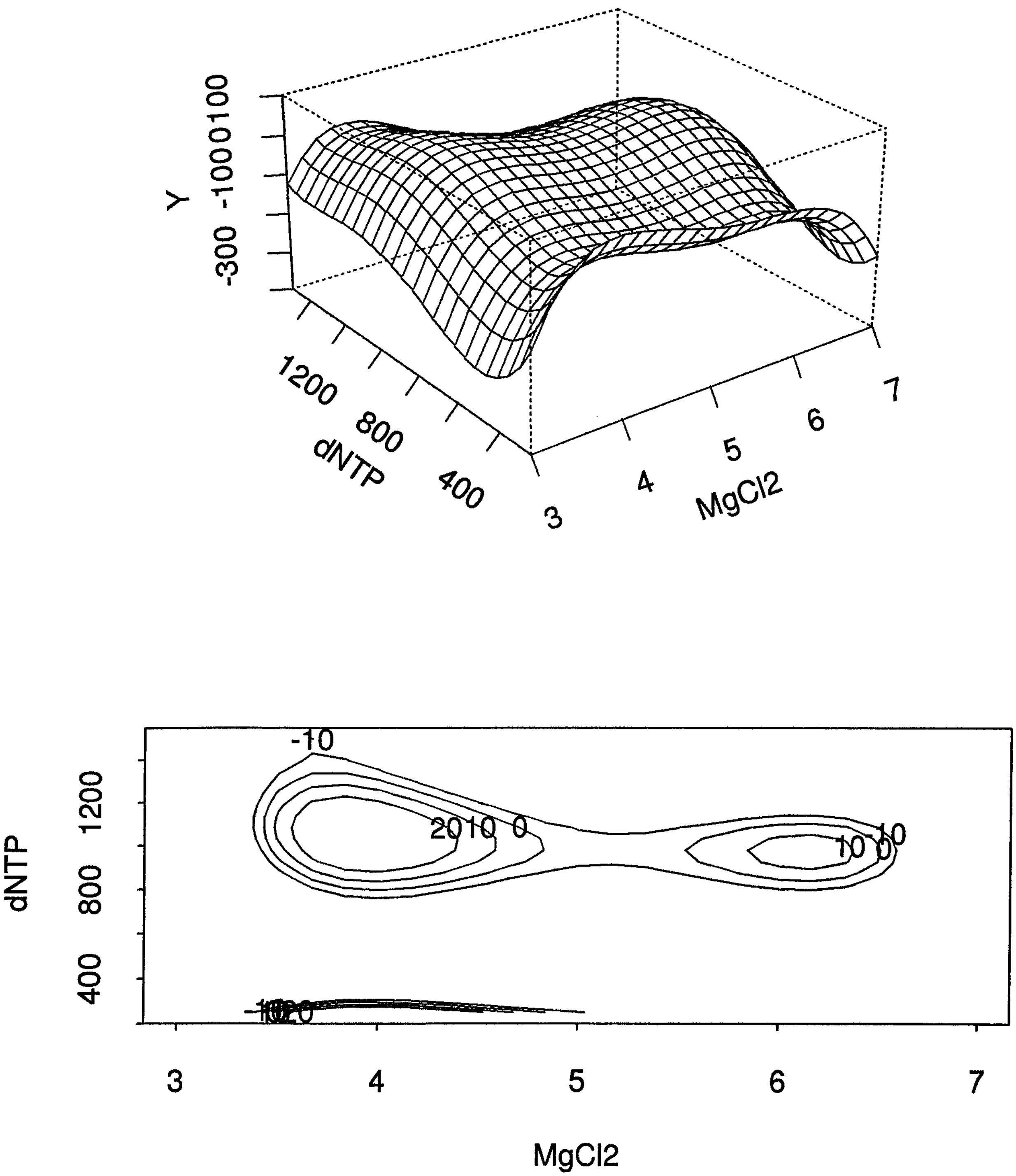


Figure 4. Response Surface for Polynomial Model



Finding the Observed Information when Using Monte Carlo E M for Mixed Models with Partially Observed / Grouped Data

Ranjini Natarajan
School of Operations Research
Cornell University
Ithaca, NY 14853

Charles E. McCulloch
Biometrics Unit and Statistics Center
Cornell University
Ithaca, NY 14853

Abstract

In this work, we develop a method to estimate the observed information matrix when using Monte Carlo E M, for a class of mixed models for partially observed/grouped data. We propose a Monte Carlo sequel to Louis' method [3]. Our method includes a Gibbs step to generate variates from the appropriate densities. We illustrate the computations involved through two examples.

1 Introduction

A computational drawback of the E M algorithm is that often the E step involves hefty, sometimes insurmountable calculations (e.g., high dimensional integration). For some problems, it may be feasible to perform these calculations using direct numerical integration [4], although for more complicated models, this might not be a computationally tractable option. Tanner [6] outlined a Monte Carlo E M algorithm, where the idea is to replace the integrals involved in the E step with a Monte Carlo estimate. We develop a Monte Carlo sequel to Louis' [3] method to estimate the observed information matrix within the M C E M framework. Although this approach works quite generally, we have worked out the details for a class of mixed models for partially observed/grouped data. By partially observed data, we refer to censored or truncated data; by grouped data we refer to ordered categorical data. Our method includes a Gibbs step to generate variates from the appropriate densities. The computations involved are illustrated through two examples.

In Section 2, we outline Louis' method and describe a Monte Carlo implementation of his method. In Section 3, we formulate the class of mixed models of interest and describe the computations involved. In Section 4, we apply the methods developed in Section 3 to probit normal regression and censored regression.

2 Louis' Method

In the usual E M terminology, we define Y to be the *latent/complete* data with probability density or mass function denoted by $[Y \mid \theta]$, where θ is the unknown parameter vector and [.] denote densities. However, we do not observe Y; instead we observe a measurable function of Y, namely, $W \sim [W \mid \theta]$. The goal of E M is to find the maximum likelihood estimate of θ based on the observed data W. The E M method is only attractive in situations where finding the complete data maximum likelihood estimator and the observed information matrix is straightforward, but the problem based on the observed data requires an iterative solution.

Define the set $\mathcal{R} = \{y : w(y) = w\}$, i.e., $\mathcal{R}$ is the set of complete data Y that could have led to the observed data W. Louis [3] proved that the observed information matrix $I_W(\theta)$ satisfies the following identity:

$$\begin{aligned} I_W(\theta) &= E(-\frac{\partial^2}{\partial\theta^2}\ln[Y \mid \theta] \mid Y \in \mathcal{R}) - \\ &\quad \text{Var}(\frac{\partial}{\partial\theta}\ln[Y \mid \theta] \mid Y \in \mathcal{R}) \qquad (1) \end{aligned}$$

The first term in $I_W(\theta)$ is simply the conditional expected information matrix of the complete data Y and is typically easy to compute. Louis proved that the second term is the expected information of the conditional distribution of Y given that Y lies in the set $\mathcal{R}$. In some applications, it may be computationally intractable to calculate the expectations in (1). Tanner [6] suggested a Monte Carlo approach to Louis' method by replacing the expectations with a Monte Carlo estimate, in the following way:

1) Generate $y_1, y_2, \ldots, y_m \sim^{iid} [Y \mid Y \in \mathcal{R}, \theta]$, for m suitably large.
2) Replace the first term in $I_W(\theta)$ by $-\frac{1}{m}\sum_{i=1}^{m}\frac{\partial^2}{\partial\theta^2}\ln[y_i \mid \theta]$ etc.

We now formulate the model of interest and illustrate the computations involved.

3 The Model

We consider the standard analysis of variance model for variance components estimation:

$$Y = X\beta + \sum_{k=1}^{r} Z_k u_k + \epsilon \quad (2)$$

$$u_k \sim N_{q_k}(0, \sigma_k^2 I) \quad (3)$$

$$\epsilon \sim N_n(0, \sigma_e^2 I) \quad (4)$$

where $Y \in \Re^{n \times 1}$ is the data vector which is partially observed or completely unobserved. $X \in \Re^{n \times p}$ is the design matrix associated with the unknown fixed effects vector $\beta \in \Re^{p \times 1}$ and $Z_k \in \Re^{n \times q_k}$ is the incidence matrix corresponding to the random effects vector u_k, $(k = 1, ..., r)$. We use the random effects structure as a convenient way to model the correlation among Y. The parameters of interest are $\theta = (\beta, \sigma_1^2, \sigma_2^2, ..., \sigma_r^2, \sigma_e^2)$.

We say a component of Y, Y_i is unobserved, if the only data information available is that it lies in some interval (a_i, b_i) where $-\infty \leq a_i < b_i \leq \infty$, and at least one of a_i, b_i is finite. Such applications arise when "experimental conditions or measuring devices permit sample points to be trapped only within specified limits" [1] as in censored or truncated data.

To put this model in the E M framework, we define the vector Y to be the complete data, since given Y, finding the maximum likelihood estimates and their standard errors is a normal linear regression problem, which is easy. We define the set $\mathcal{R} = \{Y_i : Y_i = y_i, i \in U; Y_i : a_i < Y_i < b_i, i \in C\}$ where C is the set of indices corresponding to the unobserved components of Y and U that for the observed components of Y.

The complete-data log likelihood is given by:

$$\ln[Y \mid \theta] \propto -\frac{1}{2}\ln|V| -\frac{1}{2}(Y - X\beta)' V^{-1}(Y - X\beta)$$

where $V = \sum_{k=0}^{r} \sigma_k^2 Z_k Z_k'$, $\sigma_0^2 = \sigma_e^2$ and $Z_0 = I_n$.

The first term in (1) is a matrix whose components require the calculation of expectations of the following form:

- $E((Y - X\beta))$
- $E((Y - X\beta)' V^{-1} Z_k Z_k' V^{-1} Z_l Z_l' V^{-1} (Y - X\beta))$, $k, l = 0, ..., r$

where all expectations are conditional on $Y \in \mathcal{R}$. The second term involves expectations of the following form:

- $E((Y - X\beta)(Y - X\beta)')$
- $E((Y - X\beta)' V^{-1} Z_k Z_k' V^{-1} (Y - X\beta))$, $k = 0, ..., r$
- $E((Y - X\beta)(Y - X\beta)' V^{-1} Z_k Z_k' V^{-1} (Y - X\beta))$, $k = 0, ..., r$
- $E((Y - X\beta)' V^{-1} Z_k Z_k' V^{-1} (Y - X\beta)(Y - X\beta)' V^{-1} Z_l Z_l' V^{-1} (Y - X\beta))$, $k, l, = 0, ..., r$

where all expectations are conditional on $Y \in \mathcal{R}$. So, in order to obtain a Monte Carlo estimate of $I_W(\theta)$, we need to generate $y_1, y_2, ..., y_m \sim [Y \mid Y \in \mathcal{R}, \theta]$ and then replace the expectations above by sums. It is interesting to note that we do not need to compute the first two expectations above separately, since $E((Y - X\beta)' A (Y - X\beta)) = \text{trace}(A\, E((Y - X\beta)(Y - X\beta)'))$, for any matrix A.

The density $[Y \mid Y \in \mathcal{R}, \theta]$ is not trivial to generate from, since it is the density of a multivariate normal constrained to lie within a certain set $\mathcal{R}$. We propose the use of the Gibbs sampler to generate variates from this distribution.

3.1 The Gibbs Sampler

We now outline the use of the Gibbs sampler. In order to generate a sample of Y's from the conditional distribution of $[Y \mid Y \in \mathcal{R}, \theta]$, we only need to generate the unobserved components from their full conditional distributions:

$$[Y_i, i \in C \mid Y_j, j \neq i]$$

which is a univariate truncated normal distribution, using standard results on normal theory. More formally, we have:

Step 0) Obtain starting values for Y_i, $i \in C$.

Step 1) For each $i \in C$, calculate

$$\sigma_{i \mid (i)}^2 = \text{Var}(Y_i \mid Y_j = y_j, j \neq i)$$

and the covariance $\beta_{i\,|\,(i)} = \mathrm{cov}\,(Y_i, Y_{(i)})$, where $Y_{(i)} = (Y_1, Y_2, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_n)'$.

Step 2) For each $i \in C$, calculate

$$\begin{aligned} \mu_{i\,|\,(i)} &= E(Y_i \,|\, Y_j, j \neq i) \\ &= x_i \beta + \beta'_{i\,|\,(i)} (Y_{(i)} - X_{(i)} \beta) \end{aligned}$$

where $X_{(i)} = X$ with row i deleted and x_i is the ith row of X.

Step 3) Simulate Y_i, $i \in C$ from a truncated normal distribution with mean $\mu_{i\,|\,(i)}$ and standard deviation $\sigma_{i\,|\,(i)}$, truncated between (a_i, b_i).

Repeat Steps 2 and 3 a large number of times, NREP to get $Y^{(1)}, \ldots, Y^{(NREP)}$. Discard a suitable number NBURN of the $Y^{(j)}$ from the beginning of the sequence and then retain every NSKIPth one. Ofcourse, we only need to run the Gibbs sequence one time to generate a sample from $[Y \mid Y \in \mathcal{R}, \theta]$. The advantages of this Gibbs sampling approach are two-fold. Firstly, we only ever need to generate variates from univariate truncated normal distributions, and fast acceptance-rejection algorithms exist to generate from truncated distributions [5]. Secondly, most of the computational effort is expended in repeating Steps 2 and 3 a large number of times. Thus, complicated random effects structures have little impact on the computational time, because they only affect Step 1. We verify our results on two data sets to illustrate the feasibility of the computations.

4 Examples

4.1 Probit Normal Regression

We consider a latent variable genesis of the probit normal model for binary data by postulating the existence of an underlying/latent variable Y. We assume that Y satisfies the linear mixed model in (2 - 4), with the error variance $\sigma_e^2 = 1$, without loss of generality [2]. We observe a binary variable $W_i = I(Y_i > 0)$; i.e., an indicator of whether Y crosses a threshold of 0. An example of a situation where such a threshold model might be appropriate is with regard to the *financial health* of a firm. The observed variable is an indicator of whether the firm is bankrupt (1/0), while the underlying variable represents the true health of the firm. It is unimportant whether we actually believe in the underlying variable, or merely use it as a device to estimate the parameters in the model. The advantage of this threshold model is that it automatically lends itself to a data augmentation approach such as the E M algorithm.

It is easy to see that $\mathcal{R}$ is simply the intersection of n half-lines; if $W_i = 1$, then we consider the half-line $[0, \infty)$ while if $W_i = 0$, we consider $(-\infty, 0]$. Thus, in Step 3) of the Gibbs sampler, we generate Y_i from a normal distribution, truncated *above* 0 if $W_i = 1$ and truncated *below* 0 if $W_i = 0$. We numerically verified our results on the Weil data set [7]. This data set has a treatment and control group and a single nested random effect. The response is survival status of rats and the random effect is litter. The observed data is binary indicating survival/death, and we assume it arises from a true underlying variable in the following way: $W_{ijk} = I(Y_{ijk} > 0)$ where

$$\begin{aligned} Y_{ijk} &= \beta_i + u_{ij} + \epsilon_{ijk} \\ u_{ij} &\sim N(0, \sigma_i^2 I) \\ \epsilon_{ijk} &\sim N(0, 1) \end{aligned}$$

where i indexes treatment/control, j indexes litter and k indexes the rat within the litter. So, β_i is the group mean on the latent scale and the u_{ij} are the random litter effects. The following table shows the estimates of the standard errors of the maximum likelihood estimates obtained by numerical integration (Gaussian quadrature with 20 points) and our approach.

Group		SE (M L E)	
		Numerical	*M C Louis*
Treatment	$\hat{\beta}_1$	0.309	0.304 (0.002)
	$\hat{\sigma}_1$	0.291	0.297 (0.008)
Control	$\hat{\beta}_2$	0.169	0.167 (0.007)
	$\hat{\sigma}_2$	0.301	0.302 (0.028)

The Monte Carlo estimate is the average of 35 independent runs and each run is based on a Gibbs sample of size 1500. The numbers in parenthesis are the standard errors of the Monte Carlo estimate. We can see that our estimates agree substantially with those obtained by numerical integration.

4.2 Censored Regression

We consider the case where some of the Y are right censored. This can occur when the response is a waiting time and a typical member of the population of physical or biological units is observed till an event of interest (or censoring) occurs. Such data arise in medical applications (time till the first tumor), reliability (repairable systems and software reliability) or labor economics (period of successive layoffs).

The observed data is tne pair (min(Y_i, a_i), I ($Y_i \leq a_i$), $i = 1, \dots, n$). The response vector Y is assumed to satisfy the mixed model in (2 - 4). To put this model in the E M framework, we define Y to be the complete data. It is easy to see that $\mathcal{R} = \{Y_i = y_i, i \in U, Y_i > a_i, i \in C\}$ where U is the set of indices of uncensored observations and C that for censored observations. Again, in Step 3) of the Gibbs sampler, we simply generate the censored Y_i from a normal distribution, truncated above a_i. We applied our method to a matched pairs skin graft data set analyzed by Petitt [4]. This data concerns the survival of closely and poorly matched skin grafts on the same person. The model postulated for the logarithm of of the i^{th} survival time on the j^{th} subject, denoted by Y_{ij} is:

$$\begin{aligned} Y_{ij} &= \mu + \beta_j + \gamma\, g_{ij} + \epsilon_{ij} \\ \beta_j &\sim N(0, \sigma_\beta^2) \\ \epsilon_{ij} &\sim N(0, \sigma^2) \end{aligned}$$

where β_j is a single nested individual effect, μ is the overall mean, γ is a fixed regression parameter and g_{ij} is an indicator variable (-1 for a poor match and +1 for a good match). There were 2 censored observations in this data set. We compared our results on the standard errors of the fixed effects parameters, with those obtained by Petitt and they are displayed below.

Parameter	S E (M L E)	
	Petitt	*M C Louis*
μ	0.15	0.149 (5.027e-05)
γ	0.082	0.086 (6.297e-05)

The Monte Carlo estimate is the average of 50 independent runs and each run is based on a Gibbs sample of size 2000.

5 Conclusion

In this paper, we develop a method to estimate the standard errors of the maximum likelihood estimates for a class of mixed models for incomplete data. Our approach is a valuable contribution to the existing literature on likelihood inference, since we are now able to make inferential statements in situations where it may not even be possible to compute the likelihood function with any reasonable degree of precision. In addition to the examples discussed here, we have implemented our method for the Ordinal Probit model, Tobit regression and obtained satisfactory results.

References

[1] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) "Maximum Likelihood Estimation from Incomplete Data via the E M algorithm", *J. R. Statist. Soc. B*, Vol 39, 1-38.

[2] Harville D. A., Mee R. W. (1984) "A Mixed-Model Procedure for Analyzing Ordered Categorical Data", *Biometrics*, Vol 40, 393-408.

[3] Louis, T. A. (1982) "Finding the Observed Information Matrix when Using the E M Algorithm", *J. R. Statist. Soc. B*, Vol 44, 226-233.

[4] Petitt, A. N. (1986) "Censored Observations, Repeated Measures and Mixed Effects Models: An Approach using the E M Algorithm and Normal Errors", *Biometrika*, Vol 73, 635-643.

[5] Robert C. P. (1991) "Simulation of Truncated Normal Variables", Technical Report No. 161, LSTA, University of Paris 6.

[6] Tanner M. A. (1993) "Tools for Statistical Inference", Second Edition, Springer-Verlag, NY, 1993.

[7] Weil, C. S. (1970) "Selection of the Valid Number of Sampling Units and Consideration of their Combination in Toxicological Studies Involving Reproduction, Teratogenesis, or Carcinogenesis", *Food and Cosmetic Toxicology*, Vol 8, 177-182.

A STATE-SPACE MODEL FOR LONGITUDINAL REGRESSION

Christopher H. Schmid
Biostatistics Research Center
Division of Clinical Care Research, New England Medical Center Box 63
Boston, MA 02111

ABSTRACT

The state-space representation of a regression model for longitudinal data simplifies the handling of missing data and measurement error. In this model, a continuous response depends on the lagged response and both time-dependent and time-independent covariates. The baseline response depends only on covariates. Both the EM algorithm and Gibbs sampling are used to fit the model. In EM, the E-Step uses the Kalman filter and associated filtering algorithms to update the unknown true response and predictor series for the observed data. The M-Step uses standard closed-form expressions for Gaussian data. Gibbs sampling offers a straightforward way to compute Bayesian answers and some extensions to the model.

1. INTRODUCTION

Longitudinal data, common in many scientific fields, consist of a collection of short times series taken on different units or individuals. Often, this collection of times series may be related to other collections of short series by regression. The regression model defines the conditional multivariate mean and covariance structure of the collection of response series given the predictor series and possibly fixed covariates. Since observations taken on the same individual at different times are usually correlated, the covariance structure can be quite complex.

First-order autoregressive, AR(1), models are attractive for longitudinal models with short series since they require only one correlation parameter and possess geometric rates of decay. We shall consider a form in which the response itself takes an autoregressive form as a function of the lagged response and both time-varying and time-invariant covariates. This form allows a time-varying covariate to affect all responses during and after the time at which the covariate is measured.

This model quite naturally generalizes the autoregressive time series model to include terms for trend which describe a regression on covariates. It has been called by a variety of different names in the literature including the state dependence model (Anderson and Hsaio, 1982), the conditional autoregressive model (Rosner and Munoz, 1992) and the transition (Markov) model (Zeger and Liang, 1992). Interpretation of the model parameters requires some care since the lagged response appears on the right side of the regression equation (Schmid, 1994).

Schmid, Segal, and Rosner (1994) showed how to calculate maximum likelihood estimates for this model using a Newton-Raphson algorithm when the response and the possibly time-varying covariates were subject to quantifiable random Gaussian measurement error. This implementation had several limitations, however. It involved complex analytic derivative calculations, could not handle missing values, did not provide any intuition into the sequential nature of the longitudinal data, and finally could not be easily extended to more general models.

A state-space representation of the longitudinal model treating the unobserved true series as missing data helps to rectify these problems. Considering the longitudinal regression model as a time series model with a trend component, this state-space representation can be thought of as a generalization of the one proposed by Shumway and Stoffer (1982) for time series. The state equation describes the regression model of the true series and the observation equation describes the measurement error in observing these series. Any truly missing observations in any of the observed series can be adjusted for in the observation equation. This representation leads to straightforward sequential computation by both the EM algorithm and Gibbs sampling allowing for Gaussian missing data and measurement error models. These iterative numerical algorithms can be easily extended to other models. Section 2 sets forth the state-space model, Section 3 describes the EM algorithm and Section 4 lays out the Gibbs sampler and some potential extensions of the model. Section 5 presents an application to the measurement of pulmonary function.

2. THE STATE-SPACE MODEL

A general conditional first-order autoregressive model for the outcome, y_{it}, of the ith individual at the tth time incorporating measurement error can be written

$$y_{it} = \alpha + \gamma y_{it-1} + \boldsymbol{\beta}^{\mathrm{T}} \mathbf{x}_{it} + \boldsymbol{\delta}^{\mathrm{T}} \mathbf{z}_i + \boldsymbol{\xi}^{\mathrm{T}} \mathbf{s}_{it} + \varepsilon_{it} \qquad (1)$$

where i indexes the N individuals, t indexes the T distinct times of measurement, and the ε_{it} are random errors independent for all times and individuals with a common $N(0,\sigma^2)$ distribution. Here $\mathbf{x}_{it}$ is a vector of N_X time-dependent and time-independent covariates measured without error; $\mathbf{z}_i$ is a vector of N_Z time-independent covariates measured with error; $\mathbf{s}_{it}$ is a vector of N_S time-dependent covariates measured with error at time t. The model parameters are σ^2, α, γ, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$ and $\boldsymbol{\xi}$. The first three are scalars and the last three are vectors of length N_X, N_Z and N_S, respectively.

At the baseline visit, the outcome for the ith individual, y_{i0}, is solely a function of the covariates at that visit given by

$$y_{i0} = \alpha_0 + \boldsymbol{\beta}_0^{\mathrm{T}}\mathbf{x}_{i0} + \boldsymbol{\delta}_0^{\mathrm{T}}\mathbf{z}_i + \boldsymbol{\xi}_0^{\mathrm{T}}\mathbf{s}_{i0} + \varepsilon_{i0}. \qquad (2)$$

The baseline regression parameters are σ_0^2, α_0, $\boldsymbol{\beta}$, $\boldsymbol{\delta}$ and $\boldsymbol{\xi}$ with the last three vectors of length N_X, N_Z and N_S, respectively. The ε_{i0} are random errors independent of each other and of the ε_{it} for t > 0 and follow a $N(0,\sigma_0^2)$ distribution.

Because of the measurement error and missing data, the covariates $\mathbf{z}_i$ and $\mathbf{s}_{it}$ are also stochastic quantities that we model as

$$\mathbf{s}_{it} = \boldsymbol{\alpha}_s + \boldsymbol{\gamma}_s\mathbf{s}_{it-1} + \boldsymbol{\beta}_s^{\mathrm{T}}\mathbf{x}_{it} + \boldsymbol{\delta}_s^{\mathrm{T}}\mathbf{z}_i + \boldsymbol{\varepsilon}_{sit} \qquad (3)$$

$$\mathbf{s}_{i0} = \boldsymbol{\alpha}_{s0} + \boldsymbol{\beta}_{s0}^{\mathrm{T}}\mathbf{x}_{i0} + \boldsymbol{\delta}_{s0}^{\mathrm{T}}\mathbf{z}_i + \boldsymbol{\varepsilon}_{si0} \qquad (4)$$

$$\mathbf{z}_i = \boldsymbol{\alpha}_z + \boldsymbol{\beta}_z^{\mathrm{T}}\mathbf{x}_{i0} + \boldsymbol{\varepsilon}_{zi} \qquad (5)$$

with independent errors $\boldsymbol{\varepsilon}_{sit} \sim N(0,\Sigma_s)$, $\boldsymbol{\varepsilon}_{si0} \sim N(0,\Sigma_{s_0})$ and $\boldsymbol{\varepsilon}_{z_i} \sim N(0,\Sigma_z)$. In (3) - (5), the regression parameters (with dimensions) are $\boldsymbol{\alpha}_s$ (N_s x 1), $\boldsymbol{\gamma}_s$ (N_s x N_s), $\boldsymbol{\beta}_s$ (N_s x N_x), $\boldsymbol{\delta}_s$ (N_s x N_Z), $\boldsymbol{\alpha}_{s_0}$ (N_s x 1), $\boldsymbol{\beta}_{s_0}$ (N_s x N_x), $\boldsymbol{\delta}_{s_0}$ (N_s x N_z), $\boldsymbol{\alpha}_z$ (N_z x 1), $\boldsymbol{\beta}_z$ (N_z x N_x), $\boldsymbol{\Sigma}_s$ (N_s x N_s), $\boldsymbol{\Sigma}_{s_0}$ (N_s x N_s) and $\boldsymbol{\Sigma}_z$ (N_z x N_z). Together, equations (1) - (5) may be combined into the state equations

$$\mathbf{p}_{it} = \mathbf{F}\mathbf{p}_{it-1} + \mathbf{G}\mathbf{d}_{it} + \mathbf{Q}\mathbf{e}_{it} \qquad \mathrm{t} = 1, 2, \ldots, \mathrm{T} \qquad (6)$$

and $$\mathbf{p}_{i0} = \mathbf{G}_0\mathbf{d}_{i0} + \mathbf{Q}_0\mathbf{e}_{i0} \qquad \mathrm{t} = 0 \qquad (7)$$

with $\mathbf{p}_{it} = (y_{it}\mathbf{s}_{it}\mathbf{z}_i)$, $\mathbf{d}_{it} = (1\mathbf{x}_{it})$, $\mathbf{e}_{it} = (\varepsilon_{it}\boldsymbol{\varepsilon}_{sit}) \sim N(0,\Sigma)$ and $\mathbf{e}_{i0} = (\varepsilon_{i0}\boldsymbol{\varepsilon}_{si0}\boldsymbol{\varepsilon}_{z_i}) \sim N(0,\Sigma_0)$. **F** ($1+N_S+N_Z$) x ($1+N_S+N_Z$), **G** ($1+N_S+N_Z$) x ($1+N_X$), **Q** ($1+N_S+N_Z$) x ($1+N_S$), $\mathbf{G}_0$ ($1+N_S+N_Z$) x ($1+N_X$) and $\mathbf{Q}_0$ ($1+N_S+N_Z$) x ($1+N_S+N_Z$) are transition matrices derived from equations (1) - (5). It is worth emphasizing that equations (1) - (5) describe an autoregressive process based on true rather than observed data.

To complete the state-space model, we relate the observed series $\mathbf{p}_{it}^*$ to the unobserved series $\mathbf{p}_{it}$ by the observation equations

$$\mathbf{p}_{it}^* = \boldsymbol{\Lambda}\mathbf{p}_{it} + \boldsymbol{\Phi}\mathbf{d}_{it} + \boldsymbol{\Omega}_{it} \qquad \mathrm{t} = 1, 2, \ldots, \mathrm{T} \qquad (8)$$

and $$\mathbf{p}_{i0}^* = \boldsymbol{\Lambda}_0\mathbf{p}_{i0} + \boldsymbol{\Phi}_0\mathbf{d}_{i0} + \boldsymbol{\Omega}_{i0} \qquad \mathrm{t} = 0 \qquad (9)$$

where $\mathbf{p}_{it}^* = (y_{it}^*\mathbf{s}_{it}^*)$, $\mathbf{p}_{i0}^* = (y_{i0}^*\mathbf{s}_{i0}^*\mathbf{z}_i^*)$, $\boldsymbol{\Omega}_{it} \sim N(\mathbf{0},\Sigma_\Omega)$ and $\boldsymbol{\Omega}_{i0} \sim N(\mathbf{0},\Sigma_{\Omega_0})$. $\boldsymbol{\Lambda}$ ($1+N_S$) x ($1+N_S+N_Z$), $\boldsymbol{\Phi}$ ($1+N_S$) x ($1+N_X$), $\boldsymbol{\Lambda}_0$ ($1+N_S+N_Z$) x ($1+N_S+N_Z$) and $\boldsymbol{\Phi}_0$ ($1+N_S+N_Z$) x ($1+N_X$) are the transition matrices relating the observed and true series.

The observation equations (8) and (9) describe a systematic measurement error model with the observed values related to the true values by a linear regression. The regression errors $\boldsymbol{\Omega}_{it}$ may be correlated across covariates for the same individual, but are independent across individuals. The random measurement error model is a special case with all elements of the transition matrices zero except the diagonal elements of $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}_0$.

We shall assume that the transition matrices and covariance matrices Σ_Ω and Σ_{Ω_0} in the observation equations are known. Otherwise, they may be estimated if multiple observed data series are available. When multiple measurements are unavailable, we can investigate the effect of different measurement error models through sensitivity analyses or by incorporating information from some external data source, possibly by averaging over a given measurement error distribution (Schmid and Rosner, 1993).

3. FITTING BY THE EM ALGORITHM

The EM algorithm maximizes the expected complete-data likelihood where the expectation is taken with respect to the distribution of the missing data. In this problem, the complete data consist of the observed series $\mathbf{p}_{it}^*$ and $\mathbf{x}_{it}$ and the unobserved true series $\mathbf{p}_{it}$. To simplify notation for writing the Gaussian complete-data log likelihood, express the right-hand sides of equations (1) - (5) as, respectively, $\boldsymbol{\theta}_y\mathbf{H}_{y_i}$, $\boldsymbol{\theta}_{y0}\mathbf{H}_{y0_i}$, $\boldsymbol{\theta}_s\mathbf{H}_{s_i}$, $\boldsymbol{\theta}_{s0}\mathbf{H}_{s0_i}$ and $\boldsymbol{\theta}_z\mathbf{H}_{z_i}$. In these expressions, the $\boldsymbol{\theta}$'s represent the model parameters and the **H**'s represent the model covariates. Twice the negative log likelihood is then written

$$\mathrm{N}(\log\sigma_0^2 + \mathrm{T}\log\sigma^2 + \log|\Sigma_{S_0}| + \mathrm{T}\log|\Sigma_S| + \log|\Sigma_Z|)$$

$$+\sum_{i=1}^{N}\{(y_{i0} - \boldsymbol{\theta}_{y0}\mathbf{H}_{y0_i})^{\mathrm{T}}(y_{i0} - \boldsymbol{\theta}_{y0}\mathbf{H}_{y0_i})/\sigma_0^2$$

$$+(\mathbf{s}_{i0} - \boldsymbol{\theta}_{s0}\mathbf{H}_{s0_i})^{\mathrm{T}}\Sigma_{s0}^{-1}(\mathbf{s}_{i0} - \boldsymbol{\theta}_{s0}\mathbf{H}_{s0_i})$$

$$+(\mathbf{z}_i - \boldsymbol{\theta}_z\mathbf{H}_{z_i})^{\mathrm{T}}\Sigma_z^{-1}(\mathbf{z}_i - \boldsymbol{\theta}_z\mathbf{H}_{z_i})$$

$$+\sum_{t=1}^{T}(y_{it}-\theta_y\mathbf{H}_{y_i})^{\mathrm{T}}(y_{it}-\theta_y\mathbf{H}_{y_i})/\sigma^2$$

$$+\sum_{t=1}^{T}(\mathbf{s}_{it}-\theta_s\mathbf{H}_{s_i})^{\mathrm{T}}\Sigma_s^{-1}(\mathbf{s}_{it}-\theta_s\mathbf{H}_{s_i})$$

$$+(\mathbf{p}_{i0}^{*}-\Lambda_0\mathbf{p}_{i0}-\Phi_0\mathbf{d}_{i0})^{\mathrm{T}}\Sigma_{\Omega_0}^{-1}(\mathbf{p}_{i0}^{*}-\Lambda_0\mathbf{p}_{i0}-\Phi_0\mathbf{d}_{i0})$$

$$+\sum_{t=1}^{T}(\mathbf{p}_{it}^{*}-\Lambda\mathbf{p}_{it}-\Phi\mathbf{d}_{it})^{\mathrm{T}}\Sigma_{\Omega}^{-1}(\mathbf{p}_{it}^{*}-\Lambda\mathbf{p}_{it}-\Phi\mathbf{d}_{it})\} \tag{10}$$

Assuming known parameters in the observation equation, the parts of the log likelihood above coming from the observation equation are absorbed into the constant, so that the expectation of the log likelihood with respect to the missing data distribution is proportional to the first six lines of (10).

The sufficient statistics when the expectation of this complete-data log likelihood is taken with respect to the missing data distribution are then the conditional first and second moments of the state vector $\mathbf{p}_{it}$ given the observed data. For example, $E(\mathbf{p}_{i0|T})$ represents the conditional expectation of the state vector at time 0 given all the observed data (i.e., through time T). Schmid (1994) provides the details of these expressions.

The E-Step calculates the conditional means, variances and covariances of $\mathbf{p}_{it}$ given the observed data involved in these sufficient statistics by applying the Kalman filter (Brown and Schmid, 1994; Meinhold and Singpurwalla, 1983), fixed interval smoothing algorithm (Ansley and Kohn, 1982) and state-space covariance algorithm (DeJong and MacKinnon, 1988) to each individual in the study. First, the Kalman filter sequentially computes the conditional moments of each $\mathbf{p}_{it}$ given the observed data through time t, e.g., $E(\mathbf{p}_{it|t})$, as

$$E(\mathbf{p}_{it|t-1})=\mathbf{F}E(\mathbf{p}_{it-1|t-1})+\mathbf{G}\mathbf{d}_{it}$$

$$V(\mathbf{p}_{it|t-1})=\mathbf{F}V(\mathbf{p}_{it-1|t-1})\mathbf{F}^{\mathrm{T}}+\mathbf{Q}V(\mathbf{e}_{it})\mathbf{Q}^{\mathrm{T}}$$

$$\mathbf{K}_t=V(\mathbf{p}_{it|t-1})\Lambda^{\mathrm{T}}[\Lambda V(\mathbf{p}_{it|t-1})\Lambda^{\mathrm{T}}+\Sigma_\Omega]^{-}$$

$$E(\mathbf{p}_{it|t})=E(\mathbf{p}_{it|t-1})+\mathbf{K}_t[\mathbf{p}_{it}^{*}-\Lambda E(\mathbf{p}_{it|t-1})-\Phi\mathbf{d}_{it}]$$

$$V(\mathbf{p}_{it|t})=V(\mathbf{p}_{it|t-1})-\mathbf{K}_t\Lambda V(\mathbf{p}_{it|t-1}).$$

where $V(\mathbf{e}_{it})$ is a block diagonal matrix having elements $\{\sigma^2,\Sigma_s\}$. The generalized inverse is necessary in the computation of $\mathbf{K}_t$ because rows and columns of the matrix corresponding to $\mathbf{z}_i$ will be all zeroes for t > 0. To initialize the filter, set $E(\mathbf{p}_{i0})=\mathbf{G}_0\mathbf{d}_{i0}$ and $V(\mathbf{p}_{i0})=\mathbf{Q}_0V(\mathbf{e}_{i0})\mathbf{Q}_0^{\mathrm{T}}$ with $V(\mathbf{e}_{i0})$ a block diagonal matrix having elements $\{\sigma_0^2,\Sigma_{s_0},\Sigma_z\}$. The final forward step gives the correct expectation and covariance for $\mathbf{p}_{iT}$, but the estimated moments for $\mathbf{p}_{it}$ for t < T are incompletely updated, using only data up to time t.

To complete the E-Step, work backward with the fixed interval smoothing and covariance algorithms from time T to time 0, updating the moments of $\mathbf{p}_{it}$ for $\mathbf{p}_{iu}^{*}$ and $\mathbf{x}_{iu}$ when $u>t$ by

$$\mathbf{J}_{it-1}=V(\mathbf{p}_{it-1|t-1})\mathbf{F}^{\mathrm{T}}V(\mathbf{p}_{it|t-1})^{-1}$$

$$E(\mathbf{p}_{it-1|T})=E(\mathbf{p}_{it-1|t-1})+\mathbf{J}_{t-1}[E(\mathbf{p}_{it|T})-E(\mathbf{p}_{it|t-1})]$$
$$V(\mathbf{p}_{it-1|T})=V(\mathbf{p}_{it-1|t-1})$$
$$+\mathbf{J}_{it-1}[V(\mathbf{p}_{it|T})-V(\mathbf{p}_{it|t-1})]\mathbf{J}_{it-1}^{\mathrm{T}}$$

and

$$Cov(\mathbf{p}_{it-1},\mathbf{p}_{it|T})=\mathbf{J}_{it-1}\mathrm{V}(\mathbf{p}_{it|T})$$

Each calculation in the backward step requires only output from the previous step of the backward filter and the tth step of the forward filter.

Application of standard Gaussian techniques in the M-Step then gives maximum likelihood estimates. Again, details may be found in Schmid (1994).

When values in the observed series are missing, the corresponding values in the true series become unknown even if they are not measured with error. Hence, any series with missing values must be part of $\mathbf{p}_{it}$. If the series has no observational error, then the appropriate elements of Σ_Ω are set to zero. In the filter algorithms, both the missing values in $\mathbf{p}_{it}^{*}$ and the corresponding rows of Λ and Λ_0 are set to zero. This gives the proper estimates assuming that the data are missing at random (Shumway and Stoffer, 1982). Because the state vector consists of conditionally Gaussian random variables, missing data on discrete or other non-Gaussian variables cannot be handled by these algorithms. Further details of this EM algorithm may be found in Schmid (1994).

4. FITTING BY GIBBS SAMPLING

Gibbs sampling has become a popular tool for numerically computing the posterior distribution in Bayesian models. It works by sequentially drawing from the conditional distribution of each random variable given the latest drawn values from all the other random variables in the model. In a problem with complete data **Y** and parameters θ, the algorithm involves iteratively drawing from the distribution of $\mathbf{Y}|\theta$ and then from $\theta|\mathbf{Y}$ until convergence is achieved. The final drawn values will be from the correct conditional distributions under some general conditions (Geman and Geman, 1984). The correct implementation of the Gibbs algorithm requires knowing: 1) the conditional distributions $\mathbf{Y}|\theta$ and $\theta|\mathbf{Y}$; 2) how to draw from these distributions; and 3) how to assess convergence. We shall not address the third point here but refer the reader to the literature (e.g., Gelman and Rubin, 1992; Roberts, 1992) .

In this longitudinal regression model, $\mathbf{Y}=\{\mathbf{p}_{it}, \mathbf{p}^*_{it}, \mathbf{x}_{it}\}$ for all i, t and $\theta=\{\theta_y, \theta_{y0}, \theta_s, \theta_{s0}, \theta_z, \sigma^2, \sigma_0^2, \Sigma_s, \Sigma_{s0}, \Sigma_z\}$. Using a suitable prior distribution for θ, Gibbs sampling then involves calculating the distributions of (1) $\mathbf{p}_{it}|\mathbf{p}_{it-1}, \mathbf{p}_{it+1}, \mathbf{p}^*_{it}, \mathbf{x}_{it}, \mathbf{x}_{it+1}, \theta$, (2) $\mathbf{p}^*_{it}|\mathbf{p}_{it}, \mathbf{x}_{it}, \theta$ and (3) $\theta|\mathbf{p},\mathbf{p}^*,\mathbf{x}$ where **p**, **p*** and **x** represent the collection of all $\mathbf{p}_{it}$, $\mathbf{p}^*_{it}$ and $\mathbf{x}_{it}$, respectively. Working directly with the distributions of $\mathbf{p}_{it}$ and $\mathbf{p}^*_{it}$ is more efficient than working with the distributions of the individual components of $\mathbf{p}_{it}$ that follow from equations (1)-(5). By collecting all terms involving $\mathbf{p}_{it}$ in the likelihood, we can show the distribution of $\mathbf{p}_{it}|\mathbf{p}_{it-1}, \mathbf{p}_{it+1}, \mathbf{p}^*_{it}, \mathbf{x}_{it}, \mathbf{x}_{it+1}, \theta$ to be N(**Bb**, **B**) where

$$\mathbf{B}=\left[\Sigma_{\mathbf{Q}_0}^{-1}+\mathbf{F}^{\mathrm{T}}\Sigma_{\mathbf{Q}}^{-1}\mathbf{F}+\Lambda_0^{\mathrm{T}}(\psi_0\Sigma_{\Omega_0}\psi_0^{\mathrm{T}})^{-1}\Lambda_0\right]^{-1}$$

$$\mathbf{b}=\Sigma_{\mathbf{Q}_0}^{-1}\mathbf{G}_0\mathbf{d}_{i0}+\mathbf{F}^{\mathrm{T}}\Sigma_{\mathbf{Q}}^{-1}(\mathbf{p}_{i1}-\mathbf{G}\mathbf{d}_{i1})$$

$$+\Lambda_0^{\mathrm{T}}(\psi_0\Sigma_{\Omega_0}\psi_0^{\mathrm{T}})^{-1}(\mathbf{p}^*_{i0}-\Phi_0\mathbf{d}_{i0}) \qquad \text{if } t=0;$$

$$\mathbf{B}=[\Sigma_{\mathbf{Q}}^{-1}+\mathbf{F}^{\mathrm{T}}\Sigma_{\mathbf{Q}}^{-1}\mathbf{F}+\Lambda^{\mathrm{T}}(\psi\Sigma_{\Omega}\psi^{\mathrm{T}})^{-1}\Lambda]^{-1}$$

$$\mathbf{b}=\Sigma_{\mathbf{Q}}^{-1}(\mathbf{F}\mathbf{p}_{it-1}-\mathbf{G}\mathbf{d}_{it})+\mathbf{F}^{\mathrm{T}}\Sigma_{\mathbf{Q}}^{-1}(\mathbf{p}_{it+1}-\mathbf{G}\mathbf{d}_{it+1})$$

$$+\Lambda^{\mathrm{T}}(\psi\Sigma_{\Omega}\psi^{\mathrm{T}})^{-1}(\mathbf{p}^*_{it}-\Phi\mathbf{d}_{it}) \qquad \text{if } t=1, 2, \ldots, T\text{-}1;$$

and

$$\mathbf{B}=[\Sigma_{\mathbf{Q}}^{-1}+\Lambda^{\mathrm{T}}(\psi\Sigma_{\Omega}\psi^{\mathrm{T}})^{-1}\Lambda]^{-1}$$

$$\mathbf{b}=\Sigma_{\mathbf{Q}}^{-1}(\mathbf{F}\mathbf{p}_{iT-1}-\mathbf{G}\mathbf{d}_{iT})+\Lambda^{\mathrm{T}}(\psi\Sigma_{\Omega}\psi^{\mathrm{T}})^{-1}(\mathbf{p}^*_{iT}-\Phi\mathbf{d}_{iT}) \qquad \text{if } t=T.$$

Likewise, the distribution of $\mathbf{p}^*_{i0}|\mathbf{p}_{i0}, \mathbf{x}_{i0}, \theta$ for missing values follows a normal distribution with mean $\Lambda_0\mathbf{p}_{i0}+\Phi_0\mathbf{d}_{i0}$ and variance Σ_{Ω_0}, while that for $\mathbf{p}^*_{it}|\mathbf{p}_{it}, \mathbf{x}_{it}, \theta$ is normal with mean $\Lambda\mathbf{p}_{it}+\Phi\mathbf{d}_{it}$ and variance Σ_{Ω}.

Under the standard noninformative (constant) prior,

$$\theta_y|\mathbf{p},\mathbf{p}^*,\mathbf{x},\sigma^2 \sim N[(\mathbf{H}_y{}^{\mathrm{T}}\mathbf{H}_y)^{-1}(\mathbf{H}_y{}^{\mathrm{T}}\mathbf{y}),\sigma^2(\mathbf{H}_y{}^{\mathrm{T}}\mathbf{H}_y)^{-1}]$$

$$\theta_{y0}|\mathbf{p},\mathbf{p}^*,\mathbf{x},\sigma_0^2 \sim N[(\mathbf{H}_{y0}{}^{\mathrm{T}}\mathbf{H}_{y0})^{-1}(\mathbf{H}_{y0}{}^{\mathrm{T}}\mathbf{y}_0),\sigma_0^2(\mathbf{H}_{y0}{}^{\mathrm{T}}\mathbf{H}_{y0})^{-1}]$$

$$\theta_s|\mathbf{p},\mathbf{p}^*,\mathbf{x},\Sigma_s \sim N[(\mathbf{H}_s{}^{\mathrm{T}}\Sigma_s^{-1}\mathbf{H}_s)^{-1}(\mathbf{H}_s{}^{\mathrm{T}}\Sigma_s^{-1}\mathbf{s}),(\mathbf{H}_s{}^{\mathrm{T}}\Sigma_s^{-1}\mathbf{H}_s)^{-1}]$$

$$\theta_{s0}|\mathbf{p},\mathbf{p}^*,\mathbf{x},\Sigma_{s0} \sim N[(\mathbf{H}_{s0}{}^{\mathrm{T}}\Sigma_{s0}^{-1}\mathbf{H}_{s0})^{-1}(\mathbf{H}_{s0}{}^{\mathrm{T}}\Sigma_{s0}^{-1}\mathbf{s}_0),(\mathbf{H}_{s0}{}^{\mathrm{T}}\Sigma_{s0}^{-1}\mathbf{H}_{s0})^{-1}]$$

$$\theta_z|\mathbf{p},\mathbf{p}^*,\mathbf{x},\Sigma_z \sim N[(\mathbf{H}_z{}^{\mathrm{T}}\Sigma_z^{-1}\mathbf{H}_z)^{-1}(\mathbf{H}_z{}^{\mathrm{T}}\Sigma_z^{-1}\mathbf{z}),(\mathbf{H}_z{}^{\mathrm{T}}\Sigma_z^{-1}\mathbf{H}_z)^{-1}]$$

where $\mathbf{y}$, $\mathbf{y}_0$, $\mathbf{s}$, $\mathbf{s}_0$, $\mathbf{z}$, $\mathbf{H}_y$, $\mathbf{H}_{y_0}$, $\mathbf{H}_s$, $\mathbf{H}_{s_0}$, and $\mathbf{H}_z$, are formed by stacking their respective elements.

The posterior distributions of $\sigma^2|\mathbf{p},\mathbf{p}^*,\mathbf{x}$ and $\sigma_0^2|\mathbf{p},\mathbf{p}^*,\mathbf{x}$ are inverse chi-square distributions under the standard noninformative prior for variances and those of $\Sigma_s|\mathbf{p},\mathbf{p}^*,\mathbf{x}$, $\Sigma_{s0}|\mathbf{p},\mathbf{p}^*,\mathbf{x}$ and $\Sigma_z|\mathbf{p},\mathbf{p}^*,\mathbf{x}$, are inverse Wishart distributions (Box and Tiao, 1973). The Gibbs sampler then consists of repeated sequential draws from these conditional Gaussian and Wishart distributions.

The flexibility of Gibbs sampling can facilitate computation in extensions of this model. One extension incorporates between-individual variability not captured by the regression covariates by letting the regression intercepts be random effects varying across individuals. Another uses higher-order autoregressive and moving average terms in an ARMA structure with trend given by covariates. Non-Gaussian errors and non-linear terms are other possible extensions (Carlin, Polson and Stoffer, 1992)

5. EXAMPLE

Using EM, the model was successfully applied to fit six years of pulmonary function measurements on 158 children in the Childhood Respiratory Disease Study (Redline, Tager, Segal, Gold, Speizer, and Weiss 1989) despite a substantial number of missing observations. The response forced vital capacity (FVC), the greatest volume of air a subject can forcefully expel in 6 seconds from total lung expansion, was expressed as a function of a child's age, sex, height and airways response to a cold air challenge. A total of 38 percent of the airways response, 7 percent of the height and 12 percent of the FVC measurements were missing. Details of the analysis may be found in Schmid (1994) which also describes an analysis adjusting for measurement error in FVC and airways response that was measured externally (Redline, Tager, Speizer, Rosner, and Weiss 1989).

REFERENCES

Anderson TW and Hsaio C (1982). "Formulation and estimation of dynamic models using panel data," *Journal of Econometrics,* **18**, 47-82.

Ansley GF and Kohn R (1982). "A Geometrical Derivation of the FIS Algorithm," *Biometrika,* **69**, 486-487.

Box GEP and Tiao GC (1973). Bayesian Inference in Statistical Analysis. Reading, MA: Addison-Welsey.

Brown EN and Schmid CH. (1994). "Application of the Kalman Filter to the Analysis of Biological Data" in Numerical Computer Methods, Part B: A Volume for Methods in Enzymology ed. ML Johnson and L Brand. NY: Academic Press, in press.

Carlin BP, Polson NG and Stoffer DS (1992). "A Monte-Carlo Approach to Nonnormal and Nonlinear State-Space Modeling," *J. Am. Stat. Assoc.* **87**: 493-500.

DeJong P and MacKinnon MJ (1988). "Covariances for Smoothed Estimates in State Space Models," *Biometrika* **75**, 601-602.

Gelman A and Rubin DB (1992). "A Single Series from the Gibbs Sampler Provides a False Sense of Security," in Bayesian Statistics 4 (eds. J.M. Bernardo, J. Berger, A.P. Dawid and A.F.M. Smith). Oxford: Oxford University Press, 627-633.

Geman S and Geman D (1984). "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Pattn Anal. Mach. Intell.*, **6**, 721-741.

Meinhold RJ and Singpurwalla ND (1983). "Understanding the Kalman Filter," *American Statistician,* **37**, 123-127.

Redline S, Tager IB, Speizer FE, Rosner B and Weiss ST (1989). "Longitudinal Variability in Airway Responsiveness in a Population-based Sample of Children and Young Adults," *American Review of Respiratory Disease,* **140**, 172-178.

Redline S, Tager IB, Segal MR, Gold D, Speizer FE, and Weiss ST (1989). "The Relationship Between Longitudinal Change in Pulmonary Function and Nonspecific Airway Responsiveness in Children and Young Adults," *American Review of Respiratory Disease,* **140**, 179-184.

Roberts GO (1992). "Convergence Diagnostics of the Gibbs Sampler," in Bayesian Statistics 4 (eds. J.M. Bernardo, J. Berger, A.P. Dawid and A.F.M. Smith). Oxford: Oxford University Press, 777-784.

Rosner, B and Munoz A (1992). "Conditional Linear Models for Longitudinal Data," in Statistical Models for Longitudinal Studies of Health ed. J. Dwyer, M. Feinleib, P. Lippert and H. Hoffmeister, NY: Oxford U. Press.

Schmid CH (1994). "An EM Algorithm Fitting First-Order Conditional Autoregressive Models to Longitudinal Data," *J. Am. Stat. Assoc.*, in revision.

Schmid CH and Rosner B (1993). "A Bayesian Approach to Logistic Regression Models Having Measurement Error Following a Mixture Distribution," *Statistics in Medicine* **12**, 1141-1153.

Schmid CH, Segal MR and Rosner B (1994). "Incorporating Measurement Error In the Estimation of Autoregressive Models for Longitudinal Data," *J. of Stat. Planning and Inference*, in press.

Shumway RH and Stoffer DS (1982). "An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm," *J. Time Series Analysis*, **3**, 253-264.

Zeger SL and Liang KY (1992). "An Overview of Methods for the Analysis of Longitudinal Data," *Statistics in Medicine*, **11**, 1825-1838.

REML in Generalized Linear Models: a Conditional Approach

Gordon K. Smyth
Department of Mathematics, University of Queensland, Brisbane, Q 4072, Australia

Abstract

Residual maximum likelihood estimation (REML) is often now the preferred method for estimating parameters in linear models with correlated or heteroscedastic errors. This note shows that the residual likelihood is a conditional likelihood where the conditioning is on an appropriate sufficient statistic to remove dependence on nuisance parameters. This interpretation allows a very concise derivation of the REML likelihood without the need for transformation and generalizes naturally and exactly to non-normal models in which there is a minimal sufficient statistic for the fitted values. The conditional interpretation of REML is applied to dispersion modelling in generalized linear models. It is also applied to estimate the index parameter in a power-variance family of generalized linear models.

1 Introduction

Consider the general linear model

$$\mathbf{y} = X\boldsymbol{\beta} + \mathbf{e}$$

where $\mathbf{y}$ is an $n \times 1$ vector of responses, X is an $n \times p$ design matrix of full column rank and $\mathbf{e} \sim N(0, \Omega)$ is a random vector. The variance matrix Ω is a function of a q-dimensional parameter γ, and is assumed positive definite for γ in a neighbourhood of the true value. For any given value of γ, maximum likelihood or generalized least squares lead to the estimator

$$\hat{\boldsymbol{\beta}} = (X^T\Omega^{-1}X)^{-1}X^T\Omega^{-1}\mathbf{y}$$

for $\boldsymbol{\beta}$. The problem considered in this paper is the estimation of γ.

Patterson and Thompson (1971) introduced residual maximum likelihood estimation as a method of estimating variance components in the case of unbalanced incomplete block designs. The actual derivation of the likelihood function was somewhat involved, and this prompted Harville (1974), Cooper & Thompson (1977) and Verbyla (1990) to give alternative derivations. In all of these the residual likelihood is represented as the marginal likelihood of the error constrasts. This makes generalization of the residual likelihood principle to nonlinear models or non-normal distributions difficult since zero mean error contrasts do not generally exist. The purpose of this note is to show that the residual likelihood can be viewed also as a conditional likelihood where the conditioning is on an appropriate sufficient statistic to remove dependence on the nuisance parameters. This interpretation may be of use in teaching because it clarifies the motivation for residual maximum likelihood estimation and because it allows a very concise derivation of the REML likelihood without the need for transformation of the data. It generalizes naturally and exactly to non-normal models in which there exists a minimal sufficient statistic for the fitted values.

The plan of this paper is as follows. Conditional likelihoods are discussed briefly in Section 2. The conditional derivation of REML is given in Section 3, and its generalization to generalized linear models in Section 4. Section 5 discusses dispersion estimation in generalized linear models, including the case where the dispersion is modelled using a link-linear model as in Smyth (1989). Section 6 discusses the estimation of parameters in the variance function, in a case where the exact likelihood can be specified. Emphasis in Sections 5 and 6 is given to the one-way experimental layout, since in this case the conditional likelihood can be written down in closed form. In other cases numerical evaluation or asymptotic approximation is necessary, and methods to do this are discussed also.

2 Conditional Likelihood

Consider an arbitrary likelihood function $L(\mathbf{y}; \boldsymbol{\beta}, \gamma)$ where $\boldsymbol{\beta}$ is a vector of nuisance parameters. If there exists a statistic $\mathbf{t}(\mathbf{y}; \gamma)$, possibly depending on γ, that is sufficient for $\boldsymbol{\beta}$ then the nuisance parameters can be eliminated from the likelihood by conditioning on $\mathbf{t}$. If the maximum likelihood estimation of $\boldsymbol{\beta}$ is a one-to-one function of $\mathbf{t}$, then it can be argued that there is no available information in $\mathbf{t}$ about γ in the absence of knowledge of $\boldsymbol{\beta}$, i.e., the information in $\mathbf{t}$ is entirely consumed in estimating $\boldsymbol{\beta}$. Therefore there should be no information loss in the conditional approach. The parameter of interest, $\boldsymbol{\gamma}$, can be estimated by maximizing the conditional log-likelihood $\ell_{y|t}(\mathbf{y};\boldsymbol{\gamma}) = \ell_y(\mathbf{y};\boldsymbol{\beta},\boldsymbol{\gamma}) - \ell_t(\mathbf{y};\boldsymbol{\beta},\boldsymbol{\gamma})$ which does not depend on $\boldsymbol{\beta}$.

The idea of conditioning to remove nuisance parameters is an old one (Bartlett, 1936, 1937). Kalbleisch and Sprott (1970) give an extensive discussion including the case in which $\mathbf{t}$ depends on $\boldsymbol{\gamma}$. General expressions for approximate conditional likelihoods based on saddle point approximations have been developed by Barndorff-Nielsen (1983) and Cox and Reid (1987). A long chain of related work is referenced in Cox and Reid (1987) and McCullagh and Nelder (1989, Chapter 7). Specific application to generalized linear models in made by Davison (1988).

3 A Conditional Derivation

Let $\mathbf{y}$ and $\hat{\boldsymbol{\beta}}$ be as in Section 1. For any Ω, $\hat{\boldsymbol{\beta}}$ is complete and minimal sufficient for $\boldsymbol{\beta}$, so we can eliminate $\boldsymbol{\beta}$ from the likelihood by conditioning on $\hat{\boldsymbol{\beta}}$. Since $\hat{\boldsymbol{\beta}} \sim N[\boldsymbol{\beta},(X^T\Omega^{-1}X)^{-1}]$, the conditional log-likelihood is $\ell_{y|\hat{\beta}}(\mathbf{y};\boldsymbol{\gamma}) = \ell_y(\mathbf{y};\boldsymbol{\beta},\boldsymbol{\gamma}) - \ell_{\hat{\beta}}(\mathbf{y};\boldsymbol{\beta},\boldsymbol{\gamma}) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\Omega| - \frac{1}{2}(\mathbf{y} - X\boldsymbol{\beta})^T\Omega^{-1}(\mathbf{y} - X\boldsymbol{\beta}) + \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|X^T\Omega^{-1}X| + \frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T X^T\Omega^{-1}X(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \frac{n-p}{2}\log(2\pi) - \frac{1}{2}\log|\Omega| - \frac{1}{2}\log|X^T\Omega^{-1}X| - \frac{1}{2}\mathbf{y}^T P\mathbf{y}$ where $P = \Omega^{-1} - \Omega^{-1}X(X^T\Omega^{-1}X)^{-1}X^T\Omega^{-1}$. This differs from the likelihood function given by Harville (1974) and Cooper and Thompson (1977) only in that it lacks the constant Jacobian term, $-\frac{1}{2}\log|X^TX|$, since no transformation of the data has been used.

That the conditional likelihood is equivalent to the marginal distribution of the error contrasts can be seen by transforming $\mathbf{y}$ to $\hat{\boldsymbol{\beta}}$ and $\mathbf{y}_2 = L^T\mathbf{y}$ where L is a $n\times(n-p)$ matrix of full column rank satisfying $L^TX = 0$. Conditionally, $\hat{\boldsymbol{\beta}}$ is constant, so maximizing the conditional likelihood of $\mathbf{y}$ is equivalent to maximizing the conditional likelihood of $\mathbf{y}_2$. Furthermore, $\mathbf{y}_2$ and $\hat{\boldsymbol{\beta}}$ are independent so the conditional distribution of $\mathbf{y}_2$ is the same as its marginal distribution.

In the above derivation, ℓ_y is decomposed as the sum of a marginal and a conditional likelihood. Estimation of $\boldsymbol{\gamma}$ proceeds by maximizing the conditional and then $\boldsymbol{\beta}$ is estimated by maximizing the marginal $\ell_{\hat{\beta}}$.

4 Generalized Linear Models

The generalization of REML to generalized linear models can now be stated. Consider the probability density function defined by

$$f(y;\theta,\phi) = \exp[\{y\theta - \kappa(\theta)\}/\phi + c(y,\phi)]$$

For given values of ϕ, this is a linear exponential family density function. Following Jørgensen (1987), the distribution defined by $f(y;\theta,\phi)$ is called an exponential dispersion model with dispersion parameter ϕ, and is denoted $\mathrm{ED}(\mu,\phi)$ where $\mu = E(y) = \dot{\kappa}(\theta)$. Let $y_i \sim \mathrm{ED}(\mu_i,\phi_i)$, $i = 1,\ldots,n$, be independent random variables. A generalized linear model arises if a link-linear model is assumed for the means, $g(\mu_i) = \mathbf{x}_i^T\boldsymbol{\beta}$ where $\mathbf{x}_i$ is a vector of covariates, $\boldsymbol{\beta}$ is an unknown p-vector of regression parameters and $g()$ is a known link function. We assume also that the dispersions ϕ_i depend on an unknown parameter vector $\boldsymbol{\gamma}$, for example through a link-linear model $h(\phi_i) = \mathbf{z}_i^T\boldsymbol{\gamma}$ as in Smyth (1989), where $\mathbf{z}_i$ is a vector of covariates.

Let $\Phi = \mathrm{diag}(\phi_i)$ and X be the $n \times p$ matrix with $\mathbf{x}_i^T$ as ith row. We assume $g()$ to be the canonical link function such that $g(\mu_i) = \theta_i$, so that $\mathbf{t} = X^T\Phi^{-1}\mathbf{y}$ is a complete sufficient statistic for $\boldsymbol{\beta}$. We define the REML estimate of $\boldsymbol{\gamma}$ to be that which maximizes the conditional likelihood of $\mathbf{y}$ given $\mathbf{t}$.

REML can also be used to estimate parameters in the variance function of a generalized linear model if the probability density can be completely specified. Let $\boldsymbol{\psi}$ be a parameter vector which indexes a family of exponential dispersion models, $\mathrm{ED}_{\psi}(\mu,\phi)$, and assume $y_i \sim \mathrm{ED}_{\psi}(\mu_i,\phi_i)$ with μ_i and ϕ_i as given above. In general the functions $\kappa()$, $c()$ and $g()$ will depend on $\boldsymbol{\psi}$, and $\mathrm{var}(y) = \phi_i v(\mu_i,\boldsymbol{\psi})$ where $v(\mu,\boldsymbol{\psi}) = \ddot{\kappa}(\theta)$. We define the REML estimates of $\boldsymbol{\psi}$ and $\boldsymbol{\gamma}$ to be those which maximize the conditional likelihood of $\mathbf{y}$ given $\mathbf{t}$.

The next two sections of this paper work out REML estimates for certain generalized linear models in which the conditional likelihood can be obtained in closed form.

5 Dispersion Estimation

5.1 The one-way layout

Consider a generalized linear model with means described by a one-way classification, i.e., let y_{ij}, $i = 1,\ldots,b$, $j = 1,\ldots,n_i$, be independent random variables with $y_{ij} \sim \mathrm{ED}(\beta_i,\gamma)$. The group mean $\bar{y}_i$ is sufficient for β_i and is distributed as $\mathrm{ED}(\beta_i,\gamma/n_i)$. The conditional log-likelihood is

$$\ell_{y|\hat{\beta}} = \sum_{i=1}^{b}\left\{\sum_{j=1}^{n_i}\log f(y_{ij};\theta_i,\gamma) - \log f(\bar{y}_i;\theta_i,\gamma/n_i)\right\}$$

$$= \sum_{i=1}^{b}\left\{\sum_{j=1}^{n_i} c(y_{ij},\gamma) - c(\bar{y}_i,\gamma/n_i)\right\}$$

For example suppose the y_i are normally distributed. In that case $c(y,\gamma) = -\frac{1}{2}\log\gamma - \frac{1}{2}y^2 - \frac{1}{2}\log 2\pi$ (McCullagh and Nelder, 1989), so

$$\ell_{y|\hat{\beta}} = -\frac{1}{2\gamma}D(\mathbf{y}) - \frac{N-b}{2}\log 2\pi\gamma - \frac{1}{2}\sum_{i=1}^{b}\log n_i$$

where $N = \sum n_i$ and $D(\mathbf{y}) = \sum(y_{ij} - \bar{y}_i)^2$. The conditional maximum likelihood estimator is $\hat{\gamma} = D(\mathbf{y})/(N-b)$, which is the usual residual mean square estimator of the variance in one-way analysis of variance.

If the Y_i are inverse-Gaussian, then $c(y,\gamma) = 1/(2\gamma y) - \frac{1}{2}\log\gamma - \frac{3}{2}\log y - \frac{1}{2}\log 2\pi$. In that case

$$\begin{aligned}\ell_{y|\hat{\beta}} &= -\frac{1}{2\gamma}D(\mathbf{y}) - \frac{N-b}{2}\log 2\pi\gamma \\ &\quad -\frac{3}{2}\sum_{i=1}^{b}\left(\sum_{j=1}^{n_i}\log y_{ij} - \log\bar{y}_i\right) - \frac{1}{2}\sum_{i}\log n_i\end{aligned}$$

where

$$D(\mathbf{y}) = \sum_{i=1}^{b}\sum_{i=1}^{n_i}\left(\frac{1}{y_{ij}} - \frac{1}{\bar{y}_i}\right) = \sum_{i=1}^{b}\sum_{i=1}^{n_i}\frac{(y_{ij}-\bar{y}_i)^2}{\bar{y}_i^2 y_{ij}}$$

The REML estimator of γ is the residual mean square deviance, $\hat{\gamma} = D(\mathbf{y})/(N-b)$.

In both normal and inverse-Gaussian cases, the REML estimator $\hat{\gamma}$ is uniform minimum variance unbiased for γ, and $(N-b)\hat{\gamma}/\gamma \sim \chi^2_{N-b}$ independently of the $\bar{y}_i$.

For the gamma distribution we have $c(y,\gamma) = \log(y/\gamma)/\gamma - \log y - \log\Gamma(1/\gamma)$ so

$$\begin{aligned}\ell_{y|\hat{\beta}} &= \frac{1}{\gamma}\sum_{i=1}^{b}\sum_{j=1}^{n_i}\log(Y_{ij}/\bar{Y}_i) - N\log\Gamma(1/\gamma) \\ &\quad +\sum_{i=1}^{b}\log\Gamma(n_i/\gamma) - \sum_{i=1}^{b}\left(\sum_{j=1}^{n_i}\log Y_{ij} - \log\bar{Y}_i\right)\end{aligned}$$

This is an exponential family likelihood with canonical parameter $\nu = 1/\gamma$, sufficient statistic $D(\mathbf{y}) = \sum_{i=1}^{b}\sum_{j=1}^{n_i}\log(Y_{ij}/\bar{Y}_i)$ and cumulant function $\lambda(\nu) = N\log\Gamma(\nu) - \sum_{i=1}^{b}\log\Gamma(n_i\nu)$. The REML estimator of γ is obtained by equating $D(\mathbf{y})$ to its expectation,

$$D(\mathbf{y}) = \dot{\lambda}(\nu) = N\psi(\nu) - \sum_{i=1}^{b} n_i\psi(n_i\nu)$$

where $\psi()$ is the digamma function. This can be compared to maximum likelihood estimation of γ which would have $\log(\nu)$ in place of $\psi(n_i\nu)$ in the last term. Compare with Cox and Reid (1987, p. 12) and McCullagh and Nelder (1989, p. 295).

5.2 Dispersion Modelling

Now consider the one-way layout with a link-linear model for the dispersion, i.e., suppose that the $Y_{ij} \sim \mathrm{ED}(\beta_i, \phi_{ij})$ and the ϕ_{ij} are a function of a q-vector of parameters $\boldsymbol{\gamma}$. The log-likelihood is

$$\begin{aligned}\ell_y &= \sum_{i=1}^{b}\sum_{j=1}^{n}\left\{\frac{1}{\phi_{ij}}[y_{ij}\theta_i - \kappa(\theta_i)] + c(y_{ij},\phi_{ij})\right\} \\ &= \sum_{i=1}^{b}\left\{\frac{1}{\alpha_i}[t_i\theta_i - \kappa(\theta_i)] + \sum_{j=1}^{n_i}c(y_{ij},\phi_{ij})\right\}\end{aligned}$$

where $\alpha_i = (\sum_{i=1}^{n_i}\phi_{ij}^{-1})^{-1}$, $t_i = \alpha_i\sum_{j=1}^{n_i}\phi_{ij}^{-1}y_{ij}$ and $\beta_i = \dot{\kappa}(\theta_i)$. Each t_i is sufficient for β_i and is distributed as $\mathrm{ED}(\beta_i, \alpha_i)$. The conditional log-likelihood of $\mathbf{y}$ given the t_i is

$$\ell_{y|t} = \sum_{i=1}^{b}\left\{\sum_{j=1}^{n_i}c(y_{ij},\phi_{ij}) - c(t_i,\alpha_i)\right\}$$

5.3 General Mean Models

We now leave the one-way layout and consider general link-linear models for the μ_i. Suppose that $y_i \sim \mathrm{ED}(\mu_i, \phi_i)$, $i = 1, \ldots, n$, with link-linear models for both μ_i and ϕ_i as described in Section 3. The sufficient statistic for $\boldsymbol{\beta}$ is $\mathbf{t} = X^T\Phi^{-1}\mathbf{y}$, and this has cumulant function

$$\kappa_t(\boldsymbol{\beta}) = \sum_{i=1}^{n}\phi_i^{-1}\kappa(\mathbf{x}_i^T\boldsymbol{\beta})$$

where $\kappa()$ is the cumulant function of the y_i. The cumulant generating function of t is $K(\mathbf{s}) = \kappa_t(\boldsymbol{\beta}+\mathbf{s}) - \kappa_t(\boldsymbol{\beta})$, so the probability density function of $\mathbf{t}$ is given by

$$f(\mathbf{t}) = \int\exp\left\{\sum_{i=1}^{n}\frac{\kappa(\mathbf{x}_i^T(\boldsymbol{\beta}+\mathbf{s})) - \kappa_t(\mathbf{x}_i^T\boldsymbol{\beta})}{\phi_i} - \mathbf{s}^T\mathbf{t}\right\}ds$$

The required conditional log-likelihood is

$$\ell_{y|t} = \ell_y(\mathbf{y};\boldsymbol{\beta},\boldsymbol{\gamma}) - \log f(\mathbf{t})$$

which doesn't depend on $\boldsymbol{\beta}$. Except in the normal case, the cumulant generating function of $\mathbf{t}$ is difficult to invert analytically, so either numerical evaluation or approximation will generally be necessary.

One possible approximation is to use, following a suggestion of A. T. James (James and Wiskich, 1993), the asymptotic normal approximation to the distribution of $\hat{\boldsymbol{\beta}}$. This leads to the approximate conditional log-likelihood

$$\begin{aligned}\ell_{y|\hat{\beta}} &= \ell_y(\mathbf{y};\boldsymbol{\beta},\gamma) + \frac{p}{n}\log 2\pi - \frac{1}{2}\log|X^TWX| \\ &+\frac{1}{2}(\hat{\beta}-\beta)X^TWX(\hat{\beta}-\beta)\end{aligned}$$

where $W = \mathrm{diag}\{\phi_i^{-1}v(\mu_i)\}$ and $v()$ is the variance function defined by $v(\mu) = \ddot{\kappa}(\theta)$. This expression depends on $\boldsymbol{\beta}$, but only slightly, so we can set $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}$, yielding the approximation

$$\ell_y(\mathbf{y},\hat{\boldsymbol{\beta}},\gamma) + \frac{p}{n}\log 2\pi - \frac{1}{2}\log|X^TWX| \qquad (1)$$

i.e., the log-profile likelihood for γ adjusted by the log-determinant of the covariance matrix of $\hat{\boldsymbol{\beta}}$. This method is applicable even when the link function $g()$ is not canonical, although then $\mathbf{t}$ is not sufficient so it is impossible to entirely eliminate $\boldsymbol{\beta}$ from the estimation of γ.

Another approach which leads to the same approximation in this case is to use the modified profile likelihood of Barndorff-Nielsen (1983) together with a suggestion of Cox and Reid (1987) for orthogonal parameters. The modified profile likelihood for γ is

$$\ell_y(\mathbf{y};\hat{\boldsymbol{\beta}}_\gamma,\gamma) - \frac{1}{2}\log|j_{\beta\beta}| + \log\left|\frac{\partial\hat{\beta}}{\partial\hat{\beta}_\gamma}\right|$$

where $\hat{\boldsymbol{\beta}}_\gamma$ is the maximum likelihood estimator for $\boldsymbol{\beta}$ for given γ, $\hat{\boldsymbol{\beta}}$ is the unrestricted maximum likelihood estimator, $j_{\beta\beta}$ is the observed information matrix for $\boldsymbol{\beta}$ evaluated at $\hat{\boldsymbol{\beta}}_\gamma$, and $\ell_y(\mathbf{y};\hat{\boldsymbol{\beta}}_\gamma,\gamma)$ is the log-profile likelihood for γ. Since $\boldsymbol{\beta}$ and γ are orthogonal, $\hat{\boldsymbol{\beta}}_\gamma$ varies only slowly with γ so the derivative term $\partial\hat{\beta}/\partial\hat{\beta}_\gamma$ can be neglected. For the current model we have

$$j_{\beta\beta} = X^TWX$$

and the modified profile likelihood is, apart from constants, the same as (1).

For normal linear models, the approximate conditional likelihood (1) is precisely the same as the standard residual likelihood given in Section 3. When the y_i are inverse-Gaussian and γ is scalar, modified profile likelihood leads to the residual mean deviance as the estimator of the dispersion. In other cases, the effectiveness of the approximation needs to be evaluated. This is not done here as our primary intention is to clarify the exact conditional approach.

Table 1: Simulation results for estimating γ and ϕ. One thousand data sets were generated. True values are $\gamma = 1.5$ and $\phi = 1.0$.

(a) Estimation of γ

	Mean	Std	MSE
Maximum likelihood	1.4731	0.0711	0.0058
REML	1.4873	0.0769	0.0061
Extended Quasi-Lik.	1.2345	0.0961	0.0798
Pseudo-Likelihood	1.5494	0.1894	0.0383

(b) Estimation of ϕ

	Mean	Std	MSE
Maximum likelihood	0.9010	0.1809	0.0425
REML	0.9915	0.2048	0.0420
Extended Quasi-Lik.	1.0008	0.2057	0.0423
Pseudo-Likelihood	0.9015	0.1904	0.0460

6 Variance Function Estimation

Suppose that γ is an unknown parameter than indexes a family of generalized linear models. That is, suppose that $y_i \sim \mathrm{ED}_\gamma(\mu_i,\phi)$, $i = 1,\ldots,n$ where $g(\mu_i) = \mathbf{x}_i^T\boldsymbol{\beta}$ and $\mathrm{var}(y_i) = \phi v(\mu_i,\gamma)$. The REML estimators of γ and ϕ are those which maximize the conditional likelihood of $\mathbf{y}$ given $X^T\mathbf{y}$. The purpose of this section is to consider a potentially important example, that of the compound Poisson exponential dispersion models introduced by Jørgensen (1987). The compound Poisson models have power variance functions $v(\mu,\gamma) = \mu^\gamma$ with γ between one and two. The compound Poisson distributions converge to Poisson as $\gamma \rightarrow 1$ and to gamma as $\gamma \rightarrow 2$, and so may be viewed as intermediate between the Poisson and gamma families. They are also positive and continuous except for mass at zero. Compound Poisson generalized linear models have potential applications in modelling continuous data with exact zeros, such as weather variables, insurance claims and waiting times, but the problem of estimating γ has not been satisfactorily solved (Burridge, 1987; Gilchrist, 1987).

The compound Poisson density function has been derived by Jørgensen (1992). See also Tweedie (1984). It has $\theta = \mu^{2-\gamma}/(2-\gamma)$, $\kappa(\theta) = \mu^{1-\gamma}/(1-\gamma)$ and

$$c(y,\phi) = \log\sum_{j=1}^{\infty}\frac{\{\alpha(\alpha+1)^{\alpha+1}\phi^{-\alpha-1}y^\alpha\}^j}{j!\Gamma(j\alpha)}$$

where $\alpha = (2-\gamma)/(\gamma-1)$. Tweedie (1984, p. 586) has identified $\exp c(y,\phi)$ as an instance of Wright's (1933) generalized Bessel function. It is not expressible however in terms of the more common Bessel functions.

A simulation experiment was conducted to compare four estimators of ϕ and γ. These were maximum likelihood estimation, REML, extended quasi-likelihood (Nelder and Pregibon, 1987) and pseudo-likelihood (Davidian and Carroll, 1987). Data was simulated from a one-way classification with $n_1 = \ldots = n_5 = 10$, $\beta = (0.1, 0.5, 1, 2, 5)^T$, $\phi = 1$ and $\gamma = 1.5$. One thousand such data sets were generated and, for each, γ and ϕ were estimated using the four methods. The results are tabulated in Table 1.

REML had the smallest bias for estimating γ. Maximum likelihood had the smallest standard deviation, and also the smallest mean square error, although this was not significantly different from that of REML. Pseudo-likelihood was also approximately unbiased, but with a largest standard deviation. Extended quasi-likelihood had a competitive standard deviation, but was biased down giving it the largest mean square error. Experimentation showed that the bias was due to the offset of 1/6 for zero observations. Positive and negative biases could be achieved by relatively small changes to this offset.

REML and extended quasi-likelihood were almost equally effective for estimating ϕ. The maximum likelihood estimator had again the smallest standard deviation and a mean square error not significantly greater than REML and extended quasi-likelihood, but was biased down by about 10%, as expected given the group size of ten. The pseudo-likelihood estimator was also biased down by about the same amount, despite incorporating a correction for degrees of freedom as recommended by Davidian and Carroll (1987).

We conclude that REML, in its conditional likelihood guise, is successful in reducing the bias of the maximum likelihood estimator while incurring minimal inflation to its standard deviation. Neither of its competitors, extended-quasi and pseudo likelihood, were as successful in doing this.

References

Barndorff-Nielsen, O. E. (1983). On a formula for the distribution of the maximum likelihood estimator. *Biometrika*, **70**, 343–365.

Bartlett, M. S. (1936). The information available in small samples. *Proc. Camb. Phil. Soc.*, **32**, 560–566.

Bartlett, M. S. (1937) Properties of sufficiency and statistical tests. *Proc. R. Soc.* A, **160**, 268–282.

Burridge, J. (1987). Discussion of Dr Jørgensen's paper. *J. R. Statist. Soc. B*, **49**, 150–151.

Cox, D. R. and Reid, N. (1987) Parameter orthogonality and approximate conditional inference. *J. R. Statist. Soc.* B, **49**, 1–39.

Davidian, M., and Carroll, R. J. (1987). Variance function estimation. *J. Amer. Statist. Assoc.*, **82**, 1079–1091.

Davison, A. C. (1988) Approximate conditional inference in generalized linear models. *J. R. Statist. Soc.* B, **50**, 445–461.

Gilchrist, R. (1987). Discussion of Dr Jørgensen's paper. *J. R. Statist. Soc. B*, **49**, 145–147.

James, A. T. and Wiskich, J. T. (1993). *t*-REML for robust heteroscedastic regression analysis of mitochondrial power. *Biometrics* **49**, 339–356.

Jørgensen, B. (1987). Exponential dispersion models. *J. R. Statist. Soc. B*, **49**, 127–162.

Jørgensen, B. (1992). *The theory of exponential dispersion models and analysis of deviance.* Monografias de Matemátika No. 51, Instituto de Mathemátika pura e Aplicada, Rio de Janeiro.

Kalbfleisch, J. D. and Sprott, D. A. (1970). Application of likelihood methods to models involving a large number of nuisance parameters. *J. R. Statist. Soc.* B, **32**, 175–208.

Nelder, J. A. and Pregibon, D. (1987). An extended quasi-likelihood function. *Biometrika*, **74**, 221–231.

Smyth, G. K. (1989). Generalized linear models with varying dispersion. *J. Roy. Statist. Soc.* B **51**, 47–60.

Tweedie, M. C. K. (1984). An index which distinguishes between some important exponential families. In *Statistics: Applications and New Directions. Proceedings of the Indian Statistical Institute Golden Jubilee International Conference.* (Eds. J. K. Ghosh and J. Roy), pp. 579–604. Calcutta: Indian Statistical Institute.

Verbyla, A. P. (1990). A conditional derivation of residual maximum likelihood. *Aust. J. Statist.*, **32**, 221–224.

Verbyla, A. P. (1993). Modelling variance heterogeneity: residual maximum likelihood estimation and diagnostics. *J. R. Statist. Soc.* B, **55**, 493–508.

Wright, E. M. (1933). On the coefficients of power series having essential singularities. *J. London Math. Soc.*, **8**, 71–9.

Random Integration Rules for Statistical Computation

Alan Genz [*]
Department of Mathematics
Washington State University
Pullman, WA 99164-3113
acg@eecs.wsu.edu

John Monahan
Department of Statistics
North Carolina State University
Raleigh, NC 27695-8203
monahan@stat.ncsu.edu

Abstract

Bayesian statistical applications often present high dimensional integration problems that require Monte Carlo integration. Simple Monte Carlo, in contrast to fixed integration rules, does not exploit the smoothness that one would expect from a posterior distribution. Two techniques are used to construct hybrid random multidimensional integration rules. First random orthogonal transformations are used to reduce the integration to one dimension. Then, random integration rules are derived for infinite integration intervals, generalizing rules developed by Siegel and O'Brien (1983) for finite intervals. These new rules are constructed for both Normal and Student-t weight functions. Both the combined methods produce random rules for multidimensional integrals over infinite regions with Normal or Student-t weights. Example results are presented to illustrate the effectiveness of the new rules for estimating integrals that arise in Bayesian statistical computation.

1 Introduction

A standard problem in Bayesian analysis is to numerically compute integrals in the form

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(\theta)p(\theta)d\theta_m d\theta_{m-1}\ldots d\theta_1,$$

with $\theta = (\theta_1, \theta_2, ..., \theta_m)^t$. The function $p(\theta)$ is an unnormalised posterior density function and $g(\theta)$ is some function for which an approximate expected value is needed. We will assume that the posterior density $p(\theta)$ is unimodal and approximately multivariate normal ($\theta \sim N_m(\mu, \Sigma)$, or multivariate Student-t ($\theta \sim t_m(\mu, \Sigma)$). Usually, expectations for several $g(\theta)$'s are needed, and a typical practical calculation might use a vector $\mathbf{g}(\theta) = (1, \theta, \theta\theta^t)$, so that a normalizing constant and the approximate mean and covariance matrix for θ could be determined.

This type of integration problem has traditionally been handled using Monte-Carlo algorithms. The simplest forms of these algorithms have low accuracy and slow convergence, so a number of refinements have been proposed (see the book by Davis and Rabinowitz, 1984, and the more recent paper by Evans and Swartz, 1992). One strategy that is usually effective for Monte-Carlo error reduction is importance sampling. With this strategy, $p(\theta)$ is approximated by some function $h(\theta)$, which is relatively easy to sample from. The original integral is then approximated by

$$\frac{1}{N}\sum_{i=1}^{N} g(\theta_i)(p(\theta_i)/h(\theta_i)),$$

where sample points $\{\theta_i\}$ are drawn randomly with density $h(\theta)$. The standard error from the sample provides a robust error estimate for the integral. If $h(\theta)$ is a good approximation to $p(\theta)$, then the sample variance is significantly reduced (along with the error), compared to Monte-Carlo without importance sampling.

Our new method can be considered a refinement of Monte-Carlo with importance sampling, but it should be better than simple Monte-Carlo with importance sampling because the resulting integration rule will give the exact result whenever the importance modified integrand $g(\theta)(p(\theta)/h(\theta))$ is a cubic polynomial. Simple Monte-Carlo with importance sampling results are exact whenever the importance modified integrand is constant, so the new method is expected to be significantly more accurate than simple importance sampled Monte-Carlo whenever the importance modified integrand is not constant, but still has a reasonably accurate low degree polynomial approximation.

Our method uses a multivariate normal or a multivari-

[*] Partially supported by NSF grant DMS-9211640.

ate Student-t approximation to $p(\theta)$. For these approximations, we assume that a standardizing transformation in the form $\theta = \mu + C\mathrm{x}$ has been determined for our problem, using numerical optimization if necessary. We have used μ to denote the point where $\log(p(\theta))$ is maximized, Σ to denote the inverse of the negative of the Hessian matrix for $\log(p(\theta))$ at μ, and C to denote the lower triangular Cholesky factor for Σ ($\Sigma = CC^t$). Then the transformed integrals that we consider take the form

$$I(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} w(||\mathrm{x}||)f(\mathrm{x})dx_m dx_{m-1}\ldots dx_1,$$

where $f(\mathrm{x}) = g(\mu + C\mathrm{x})p(\mu + C\mathrm{x})/w(||\mathrm{x}||)$, and $w(||\mathrm{x}||) = e^{-\mathrm{x}^t\mathrm{x}/2}$, or $w(||\mathrm{x}||) = (1 + \frac{\mathrm{x}^t\mathrm{x}}{\nu})^{-(m+\nu)/2}$. If our approximation to the posterior density is a good one, then we expect $f(\mathrm{x})$ to be well approximated by a low degree polynomial in x, and this motivates our construction of random multimensional integration rules for polynomials. These rules are generalizations of the degree three rules derived for the interval [-1,1], with weight $w(r) = 1$, developed by Siegel and O'Brien (1983). Earlier work by Hammersley and Handscomb (1964) also considered the construction of random integration rules for finite intervals.

Our development of the random multidimensional integration rules requires an additional change of variables to a radial-spherical coordinate system. We let $\mathrm{x} = r\mathrm{z}$, with $\mathrm{z}^t\mathrm{z} = 1$, so that $\mathrm{x}^t\mathrm{x} = r^2$, for $r \in [0, \infty)$. Then

$$\begin{aligned} I(f) &= \int_{\mathrm{z}^t\mathrm{z}=1}\int_0^{\infty} w(r)r^{m-1}f(r\mathrm{z})d\mathrm{z}dr \\ &= \frac{1}{2}\int_{\mathrm{z}^t\mathrm{z}=1}\int_{-\infty}^{\infty} w(r)|r|^{m-1}f(r\mathrm{z})d\mathrm{z}dr/2. \end{aligned}$$

We want to compute numerical approximations to $I(f)$ so we need integration rules for the surface of the unit m-sphere defined by $z^t z = 1$, and for the radial interval $[-\infty, \infty]$. For the spherical surface integrals we use

$$S_Q(s) = \frac{1}{2m}\sum_{j=1}^{m}(s(-Qe_j) + s(+Qe_j)) \approx \int_{z^t z=1} s(z)dz,$$

where $e_j = (0, \ldots, 0, 1, 0, \ldots, 0)^t$ and Q is an $m \times m$ random orthogonal matrix.. The integration rule $S_Q(s)$ is a degree three rule (see Stroud, 1971, p. 294) for the surface of the unit m-sphere. If Q is chosen uniformly (see Stewart, 1980), S_Q is an unbiased random degree three rule for the surface of the unit m-sphere. Deák (1990) uses a transformation to a similar spherical coordinate system with random orthogonal transformations to develop a method for computing multivariate normal probabilities. We still need random degree three rules for integrals of the form $\int_{-\infty}^{\infty}|r|^{m-1}e^{-r^2/2}h(r)dr$, or $\int_{-\infty}^{\infty}|r|^{m-1}(1 + \frac{r^2}{\nu})^{-(m+\nu)/2}h(r)dr$. We discuss these degree three radial rules in the next section. We then show how the radial rules can be combined with the spherical surface rules to produce random rules for $I(f)$. In the section three, we demonstrate the use of the new rules with two test Bayesian computation problems.

2 Random Radial Rules

We define a basic radial integration rule $R_\rho(h)$ by

$$\begin{aligned} R_\rho(h) &= h(0) + \frac{\alpha}{2\rho^2}(h(\rho) + h(-\rho) - 2h(0)) \\ &\approx \int_{-\infty}^{\infty}|r|^{m-1}w(r)h(r)dr, \end{aligned}$$

where ρ is a positive real number, $w(r)$ is now normalized so that $\int_{-\infty}^{\infty}|r|^{m-1}w(r)dr = 1$, and $\alpha = \int_{-\infty}^{\infty}|r|^{m+1}w(r)dr$. We can prove (see Genz and Monahan, 1994) the following theorem, which establishes two important properties of the rules $R_\rho(h)$.

Theorem 1 *If ρ is a random variable on $(0, \infty)$, with density $\frac{2}{\alpha}r^{m+1}w(r)$, then*

$$R_\rho(h) = \int_{-\infty}^{\infty}|r|^{m-1}w(r)h(r)dr,$$

whenever h is cubic polynomial, and

$$E\{R_\rho(h)\} = \int_{-\infty}^{\infty}|r|^{m-1}w(r)h(r)dr,$$

for any integrable h.

For the two specific weight functions that we are interested in, we have determined that $\alpha = m$ when $w(r) \sim e^{-r^2/2}$, and $\alpha = \frac{m\nu}{\nu-2}$ when $w(r) \sim (1 + \frac{r^2}{\nu})^{-(m+\nu)/2}$.

A random degree three radial rule R_ρ can now be combined with a random degree three spherical rule S_Q to produce a random degree three rule for $I(f)$. We first let $D_\rho(f, x) = (f(\rho x) + f(-\rho x) - 2f(0))/(2\rho^2)$. Then our combined random spherical radial integration rule for $I(f)$ is given by

$$\begin{aligned} SR_{Q,\rho}(f) &= \frac{1}{2m}\sum_{j=1}^{m}(\;\; (f(0) + \alpha D_\rho(f, Qe_j)) \\ &\qquad + (f(0) + \alpha D_\rho(f, -Qe_j)) \;\;) \\ &= f(0) + \frac{\alpha}{2m}\sum_{j=1}^{m} D_\rho(Qe_j). \end{aligned}$$

It is easy to establish the following (Genz and Monahan, 1994)

Theorem 2 *If ρ is a random variable on $(0,\infty)$ with density $\frac{2}{\alpha}r^{m+1}w(r)$, and Q is an $m\times m$ uniform random orthogonal matrix, then*

$$SR_{Q,\rho}(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} w(||x||)f(x)dx$$

whenever f is cubic polynomial, and

$$E\{SR_{Q,\rho}(f)\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} w(||x||)f(x)dx$$

for any integrable f.

The unbiased degree three rules $SR_{Q,\rho}(f)$ form the basis for the following algorithm:

Spherical-Radial Rule Integration Algorithm

```
1. Input $\epsilon$, $m$, $f$, $w$ and $N_{max}$.
2. Set $N = 0$, $I = 0$, $V = 0$ and compute $\alpha$ and $f(0)$.
3. Repeat
    (a) Set $SR = 0$.
    (b) Generate a uniformly random orthogonal $m \times m$ matrix $Q$.
    (c) Generate $\rho$ from the density $\frac{2}{\alpha}r^{m+1}w(r)$.
    (d) For $j = 1, 2, ..., m$ set
        $SR = SR + (f(\rho Qe_j) + f(-\rho Qe_j) - 2f(0))/(2\rho^2)$.
    (e) Set $SR = f(0) + \alpha SR/m$, $N = N + 1$,
        $D = (SR - I)/N$, $I = I + D$ and
        $V = V + (N-1)ND^2$.
  Until $\sqrt{V/(N(N-1))} < \epsilon$ or $N = N_{max}$.
4. Output $I \approx I(f)$, $\sigma = \sqrt{V/(N(N-1))}$ and $N$.
```

The input ϵ is an error tolerance, the input N_{max} provides a limit on the time for the algorithm, and the output σ is the standard error for the integral estimate I.

3 Examples

The first example is a three dimensional nonlinear regression problem from the time series book by Fuller (1976). The posterior is given by

$$p(\theta) = (10 + \sum_{i=1}^{12}(y_i - \theta_1 - \theta_2 e^{-\theta_3 t_i})^2)^{-6.1}$$

with $\theta \in (-\infty,\infty)^3$. We model $p(\theta)$ with a multivariate normal approximation, so we use

$$f(x) = Ke^{\mathbf{x}^t\mathbf{x}/2}p(\mu + C(x_1, x_2, x_3)^t),$$

after computing the mode μ and C for $\log(p)$. The constant K is chosen to prevent underflow in the numerical evaluation of f. For this example $K = e^{30}$. In the following table we show results from the SR rules. For comparison, we also show results from simple importance sampled Monte-Carlo rules, where the components for the sample points $\mathbf{x}$ are random drawn from $N(0,1)$. The entries in the error columns are the standard errors obtained from the random samples for the respective methods..

Test Results with 10,000 f Values

	Simple M-C		SR Rules	
f	$E\{f\}$	Error	$E\{f\}$	Error
p/w	0.5773	0.0403	0.5277	0.0103
$\theta_1 p/w$	140.6531	9.5358	141.1459	2.7229
$\theta_2 p/w$	-83.7048	6.0097	-83.5633	1.6003
$\theta_3 p/w$	1.4968	0.1487	1.4242	0.0334

The standard errors for the simple Monte-Carlo rules are 3-4 times larger than those for SR rules. This indicates that approximately ten times more computer time would be needed for the crude Monte-Carlo rules to achieve an accuracy level similar to that achieve by the SR rules.

For our second example we use a seven dimensional proportional hazards model problem discussed by Dellaportas and Wright (1992) and Lawless (1982). The posterior is given by

$$p(\rho,\beta) = \prod_{i=1}^{48}\rho t_i^{\rho-1}e^{\mathbf{z}_i^t\beta}\prod_{i=1}^{65}e^{t_i^\rho e^{\mathbf{z}_i^t\beta}}$$

with $\rho > 0$ and $\beta \in (-\infty,\infty)^6$. After we first transform ρ using $x_1 = \log(\rho)$, we model $p(\theta)$ with a multivariate normal approximation. So we use

$$f(x) = Ke^{\mathbf{x}^t\mathbf{x}/2}e^{x_1}p(\mu + C(e^{x_1}, x_2, ..., x_7)^t),$$

after computing the mode μ and C for $\log(\rho) + \log(p)$. For this example $K = e^{215}$. In the following table we show results for the SR rules and simple importance sampled Monte-Carlo rules.

Test Results with 75,000 f Values

	Simple M-C		*SR* Rules	
f	$E\{f\}$	Error	$E\{f\}$	Error
p/w	0.3918	0.0026	0.3907	0.0015
$\rho p/w$	1.1747	0.0086	1.1737	0.0047
$\beta_1 p/w$	-4.0611	0.0319	-4.0573	0.0169
$\beta_2 p/w$	1.9680	0.0194	1.9438	0.0065
$\beta_3 p/w$	-0.1217	0.0008	-0.1216	0.0005
$\beta_4 p/w$	-0.0192	0.0002	-0.0193	0.0001
$\beta_5 p/w$	-0.0418	0.0033	-0.0418	0.0008
$\beta_6 p/w$	0.1251	0.0013	0.1251	0.0004

For this example the *SR* rule results have standard errors that are approximately half as large as those for the simple Monte-Carlo. These results are not as good as those for the three dimensional problem, but the *SR* rules are still approximately four times more efficient than the simple Monte-Carlo rules.

In order to monitor, and possibly improve, the convergence of the *SR* rules, we have considered the development of convergence diagnostics for the simple *SR* integration algorithm described in the previous section. The integrand $f(\mathrm{x})$ for a given integration problem could have more (or less) variation around the spherical surface than it does along radial directions. If we had diagnostics to determine these differences in variation, our simple algorithm could be modified to increase sampling in either the spherical or radial directions, in order to to adapt to these differences.

A natural diagnostic for the spherical variation for a fixed radius ρ is the sample variance for the *SR* average that is accumulated in the loop at step 3(d) in the algorithm. Alternately, with Q fixed, a loop could be introduced at step 3(c) so that several different ρ's could be used and the variance in the resulting *SR* rules could be used as a diagnostic for radial variation. A relatively large variation in the radial direction might indicate that a multivariate normal model was not valid. Therefore a multivariate Student t model might be more appropriate, and/or the number of samples in the radial direction could be increased. Alternatively, a relatively large variance for the *SR* average would suggest that more Q's should be used for each ρ. This could be accomplished by interchanging steps 3(b) and 3(c) and adding a loop at the modified step 3(c) that would allow several different Q's to be generated for each ρ. A more general algorithm could have nested loops at both steps 3(b) and 3(c), with lengths dynamically adjusted to balance the radial and spherical variances.

4 Concluding Remarks

We have described degree three random integration rules that can be used to numerically estimate integrals over infinite regions. Results from two examples suggest that averages of samples of these rules can provide more accurate integral estimates than simpler Monte-Carlo importance sampling methods. However, in contrast to traditional polynomial rules for numerical integration, the standard errors from the random rule samples can be used for robust error estimation.

For future work with these rules we intend to consider more examples, and develop and implement heuristics to automate the incorporation of the variance diagnostics into our algorithm. We also hope to extend our work to include random degree five rules for infinite regions.

References

Davis, P. J. and Rabinowitz P. (1984), ***Methods of Numerical Integration***, Academic Press, New York.

Dellaportas, P. and Wright, D. (1992), A numerical Integration Strategy in Bayesian Analysis, in ***Bayesian Statistics 4***, Bernardo, J.M., Berger, J.O., David, A.P. and Smith, A.F.M. (Eds.), Oxford University Press, Oxford, pp. 601-606.

Deák, I. (1990), ***Random Number Generation and Simulation***, Akadémiai Kiadó, Budapest.

Evans, M. and Swartz, T. (1992), Some integration strategies for problems in statistical inference. *Computing Science and Statistics*, 24, 310-317.

Fuller, W.A. (1976), ***Introduction to Statistical Time Series***, John Wiley and Sons, New York, p. 228.

Genz, A. and Monohan, J. (1994), Randomized Degree Three Rules for Infinite Integration Regions, in preparation.

Hammersley, J.M. and Handscomb, D.C. (1964), ***Monte Carlo Methods***, Chapman and Hall, London.

Siegel, A.F. and O'Brien, F. (1983), Unbiased Monte Carlo Integration Methods with Exactness for Low Order Polynomials, *SIAM. J. Sci. Stat. Comput.* **6**, 169-181.

Stewart, G.W. (1980), The Efficient Generation of Random Orthogonal Matrices with An Application to Condition Estimation, *SIAM J. Numer. Anal.* **17**, 403-409.

Stroud, A. H. (1971), ***The Approximate Calculation of Multiple Integrals***, Prentice Hall, Englewood Cliffs, New Jersey.

Use PVM on Computation of Analysis of Repeated Measurement Designs

Chen-Chi Shing
Computer Science Department, Radford University, Radford, Virginia 24142

Abstract

PVM is a software that allows a heterogeneous network of parallel and serial computers to use distributed memery to do concurrent computation. Due to lack of accessing a parallel computer to do complicated computation, we show how to parallelize the Beaton's sweep operation on computation of the analysis of regression and designed experiments, therefore the analysis of repeated measurement designs under PVM tool. The performance of the parallelized sweep operator will be evaluated.

1. Introduction

Parallel processing is increasingly important in scientific fields such as statistical computing. In statistically computational intensive area such as regression analysis and analysis of experimental designs, existing sequential algorithms should be parallelized so that applications can be processed by parallel computers to speed up the computation. However, parallel computers are too costly for most colleges to be obtained.

A high level programming environment, called PVM (Parallel Virtual Machine), can be utilized in clusters of heterogeneous networked Unix workstations and parallel computers to do parallel processing without calling low level Unix utilities from communication layer such as socket([1],[4],[7]). PVM is an on going project, started in the summer of 1989 at Oak Ridge National Lab. Basically, PVM generates a series of tasks, like Unix processes. They are synchronized by using message-passing technique to pass data between them and solve a problem in parallel. The applications can be programmed in either Fortran 77 or C.

In computation of analysis of regression and designs of experiments a common method used to solve a normal equation in most statistical software is the Beaton's sweep operation ([2],[3]). In addition, this method also gives insight into the least square method. Some important statistical measures are the products of the process of sweeping. However, the sweep operation is designed to be sequential and it is difficult to utilize the power of parallel computers while sweeping.

In this paper a method is proposed to parallelize the sweep operation. The algorithm is implemented in Fortran 77 under PVM 3.1 and the speed-up is evaluated([8]).

2. Sweep Operation

The most fundamental operation in regression analysis and analysis of variance is Beaton's sweep operation. It is one of the most simple methods to solve the normal equation and also a process of matrix inversion by bordering([3]). Beaton's sweep operation sweeps through a positive semidefinite matrix, the design matrix X'X or S matrix, and produces sum of square of residuals and regression coefficients for a regression model, or sum of squares for the residuals and all effects for a model of designed experiments([6]).

The algorithm of the sweep operation is described as follows:

Sweep on i th row of the S=(S_{ij}) matrix and resulting a matrix T=(T_{ij}):

If $S_{ii} \neq 0$, then $T_{ii} = -\frac{1}{S_{ii}}$ and $T_{ij} = \frac{S_{ij}}{S_{ii}}$, for $j \neq i$. Also $T_{jk} = S_{jk} - S_{ji} \times \frac{Sik}{S_{ij}}$, for $k \neq i$ and $j \neq i$, else $T_{ii} = 0$ and $T_{ij} = 0$, for $j \neq i$. Also $T_{jk} = S_{jk}$, for $k \neq i$ and $j \neq i$.

The sweep operator has the properties of associativity and commutativity. The operation is purely designed as sequential, that is, the matrix which is used for the current sweep completely depends on the resulting matrix of the previous sweep. In order to parallelize the sweep operation so it can be used in regression analysis and analysis of experimental designs, the operation must be studied in detail.

3. Parallelize Sweep Operation

Given a n × m regression matrix X the positive semidefinite symmetric matrix S=(S_{ij}) with dimension m in Ith sweep, can be divided into four different areas as follows:

Area A: S_{ij}, i=I+1, and I+1≤ j ≤m
Area B: S_{ij}, I+2≤ i ≤m, and I+1≤ j ≤m
Area C: S_{ij}, 1≤ i ≤I, and I+1≤ j ≤m
Area D: S_{ij}, 1≤ i ≤I, and 1≤ j ≤I

Since the I+1th sweep must use the value of S_{i+1i+1}, area A and B can be performed first and the elements in area C and D during Ith sweep does not depend on the next sweep. Hence area C and D of the current sweep can be performed concurrently with area A and B of the next sweep. By overlapping the area C and D and area A and B of the problem domain, we can save part of the total sweeping time. Areas A, B, C and D will be processed concurrently by computers A, B, C and D in a network. Since matrix S is symmetric, we can only compute either upper or lower diagonal elements during sweeping in order to save execution time. Results will be distributed from computer A to B, B to C, C to D.

In the next section PVM will be used to implement the parallel concept.

4. Implementation Under PVM

PVM can be started heterogeneously in different hosts through a background daemon process called *pvmd* in each host. Users are very easy to add or delete hosts as their wishes in PVM environment. Each daemon will communicate each other through message passing. Under master-slave model it is easy to set a central control for user to do input and output. That is, the master program will allow user to enter the initial data and also collect the results to the user. Master program and each slave program will be run on each different host. And data sent as messages will be passed between master and slaves. Each host runs the same number of sweep operations except that host A will sweep on only the elements in area A. Similarly for other hosts.

The process of writing a program under PVM is a little complicated. Once user has run a sequential program successfully in one host, one will implement the corresponding parallel program with different tasks (or processes) into the same host. Finally, the successful parallel program will be implemented under different hosts.

4.1. Algorithm

Based on master-slave model, the master program will spawn four different processes, i.e. module A, B, C and D. Each module receives initial matrix information and sweeps on the corresponding area of the matrix by using message passing to maintain the order of sweeping. The sweeping oder is by executing module A first. Then the results of module A will be sent to module B. After module B finishes, the results will be sent to module C to continue the current sweep. In the mean time the results of module B will be sent to module A again to start the next sweep operation if necessary. Once module C finishes, it will send the results to module D to finish up the current sweep operation. If necessary, the results of module D of the current sweep and the results of module B of the next sweep will be sent altogether to module C of the next sweep to continue. After the number of sweeping user requested to do, the final resulting matrix will be sent back to master program.

The following are the algorithms for each program used:

```
Algorithm for master program:
1. spawn tasks module A, B, C and D.
2. enter user's data matrix S
3. pack all data
4. broadcast data to module A, B, C and D.
5. wait for receiving results from module D
6. unpack the results
7. output the results

Algorithm for module A program:
1. receive the matrix S from master program
2. unpack data
3. loop until no more sweeps
   (i) receive the last sweep from module B
   (ii) unpack results
   (iii) sweep the matrix in area A
   (iv) pack results
   (v) send the results to module B
endloop

Algorithm for module B program:
1. receive the matrix S from master program
2. unpack data
3. pack initial S data matrix
4. send initial S data matrix to start
   module A and trigger the start of the
   sweep operation
5. loop until no more sweeps
   (i) receive the results from module A
   (ii) unpack results
   (iii) sweep the matrix in area B
   (iv) pack results
   (v) send the results to module C
   (vi) send the results to module A
endloop

Algorithm for module C program:
1. receive the matrix S from master program
2. unpack data
3. loop until no more sweeps
   (i) receive the results from module B
   (ii) unpack results
   (iii) receive the last sweep from module D
   (iv) unpack results
   (v) sweep the matrix in area C
   (vi) pack results
   (vii) send the results to module D
```

```
endloop

Algorithm for module D program:
1. receive the matrix S from master program
2. unpack data
3. loop until no more sweeps
   (i) receive the results from module C
   (ii) unpack results
   (iii) sweep the matrix in area D
   (iv) pack the results
   (v) send the results to module C
endloop
4. pack the final results
5. send the results back to master program
```

4.2. Example

Given m=4 and w=3, the matrix X is

$$X = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 3 \\ 1 & 2 & 3 & -1 \\ 1 & 3 & 0 & 2 \end{pmatrix}$$

Suppose P_i denotes the initial matrix going to module P before sweeping on area P in ith sweep, then following the algorithm above, we have

$$A_1 = S = \begin{pmatrix} 4 & 7 & 6 & 5 \\ 7 & 15 & 9 & 8 \\ 6 & 9 & 14 & 4 \\ 5 & 8 & 4 & 15 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 4 & 7 & 6 & 5 \\ 7 & 2.75 & -1.5 & -0.75 \\ 6 & -1.5 & 14 & 4 \\ 5 & -0.75 & 4 & 15 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 4 & 7 & 6 & 5 \\ 7 & 2.75 & -1.5 & -0.75 \\ 6 & -1.5 & 5 & -3.5 \\ 5 & -0.75 & -3.5 & 8.75 \end{pmatrix}$$

$$B_2 = \begin{pmatrix} 4 & 7 & 6 & 5 \\ 7 & 2.75 & -1.5 & -0.75 \\ 6 & -1.5 & 4.182 & -3.909 \\ 5 & -0.75 & -3.909 & 8.75 \end{pmatrix}$$

$$C_1 = \begin{pmatrix} 4 & 7 & 6 & 5 \\ 7 & 2.75 & -1.5 & -0.75 \\ 6 & -1.5 & 5 & -3.5 \\ 5 & -0.75 & -3.5 & 8.75 \end{pmatrix}$$

$$D_1 = \begin{pmatrix} 4 & 1.75 & 1.5 & 1.25 \\ 1.75 & 2.75 & -1.5 & -0.75 \\ 1.5 & -1.5 & 5 & -3.5 \\ 1.25 & -0.75 & -3.5 & 8.75 \end{pmatrix}$$

$$C_2 = \begin{pmatrix} -0.25 & 1.75 & 1.5 & 1.25 \\ 1.75 & 2.75 & -1.5 & -0.75 \\ 1.5 & -1.5 & 4.182 & -3.909 \\ 1.25 & -0.75 & -3.909 & 8.545 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} -0.25 & 1.75 & 2.455 & 1.727 \\ 1.75 & 2.75 & -0.545 & -0.273 \\ 2.455 & -0.545 & 4.182 & -3.909 \\ 1.727 & -0.273 & -3.909 & 8.545 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} -0.25 & 1.75 & 1.5 & 1.25 \\ 1.75 & 2.75 & -1.5 & -0.75 \\ 1.5 & -1.5 & 4.182 & -3.909 \\ 1.25 & -0.75 & -3.909 & 8.545 \end{pmatrix}$$

$$B_3 = \begin{pmatrix} -0.25 & 1.75 & 1.5 & 1.25 \\ 1.75 & 2.75 & -1.5 & -0.75 \\ 1.5 & -1.5 & 4.182 & -3.909 \\ 1.25 & -0.75 & -3.909 & 4.891 \end{pmatrix}$$

$$C_3 = \begin{pmatrix} -1.364 & 0.636 & 2.455 & 1.727 \\ 0.636 & -0.364 & -0.545 & -0.273 \\ 2.455 & -0.545 & 4.182 & -3.909 \\ 1.727 & -0.273 & -3.909 & 4.891 \end{pmatrix}$$

$$D_3 = \begin{pmatrix} -1.364 & 0.636 & 2.455 & 4.02 \\ 0.636 & -0.364 & -0.545 & -0.78 \\ 2.455 & -0.545 & 4.182 & -0.935 \\ 4.02 & -0.78 & -0.935 & 4.891 \end{pmatrix}$$

And the resulting matrix is

$$\begin{pmatrix} -2.805 & 0.956 & 0.587 & 4.02 \\ 0.956 & -0.435 & -0.130 & -0.78 \\ 0.587 & -0.130 & -0.239 & -0.935 \\ 4.02 & -0.78 & -0.935 & 4.891 \end{pmatrix}$$

5. Performance Evaluation

Using the parallel algorithm used in the last section, the number of different arithmetic operations is described in the following table:

At Ith sweep:

Module	Number of +,-	Number of *,/
A	m-I	2(m-I)
B	(m-I)$\frac{(m-I-1)}{2}$	(m-I)(m-I-1)
C	(I-1)(m-I)	(m-I)(2I-1)
D	(I-1)$\frac{I}{2}$	I(I-1)+I

For Sparc Station, the time to execute a floating point * or / is about twice as much as that of + or -. Let u be the time to perform one + or - operation, then in terms of u time unit, at Ith sweep, module A, B, C and D take 5(m-I), 5(m-I)$\frac{(m-I-1)}{2}$, (m-I)(5I-3) and (5I-1)$\frac{I}{2}$ times, respectively.

Since module C of the current sweep starts at the same time as module A of the next sweep, the time which the parallel algorithm can be saved is either

$\sum(timeC + D)$ on the Ith sweep if time A+B on the (I+1)th sweep $\geq$ time C+D on Ith sweep. That is, $0 \leq I \leq \lfloor a \rfloor$ or $\lceil b \rceil \leq I \leq w$,

or $\sum(timeA + B)$ on the (I+1)th sweep otherwise. That is, $\lceil a \rceil \leq I \leq \lfloor b \rfloor$, where w is the number of sweeps user request to do,

$$a = \frac{10m - \sqrt{50m^2 - 10m}}{10}$$

and

$$b = \frac{10m + \sqrt{50m^2 - 10m}}{10}$$

with m is the dimension of the data matrix S.

Then the total time saved after w sweeps using the parallel algorithm over the sequentail algorithm in terms of time unit u is

$\frac{5}{6}[\lfloor b\rfloor * (\lfloor b\rfloor - 1) * (2\lfloor b\rfloor - 1) - \lfloor a\rfloor * (\lfloor a\rfloor + 1) * (2\lfloor a\rfloor + 1)]$ $-\frac{5}{12}$w(w-1)(2w-1) +(5m+$\frac{5}{2}$) $\frac{1}{2}$w(w-1) +5m $[\lfloor a\rfloor * (\lfloor a\rfloor + 1) - \lfloor b\rfloor * (\lfloor b\rfloor - 1)]$ -3m(w-1) +$(\frac{5}{2}m^2 + \frac{m}{2})$ $(\lfloor b\rfloor - \lfloor a\rfloor - 1)$

The total time after w sweeps for the sequential algorithm is $\frac{wm(5m-1)}{2}$.

Assuming no delaying time for the messages being sent,wait and received, then for m=200 matrix S after w=197 sweeps, $\lfloor a\rfloor = 58$ and $\lfloor b\rfloor =$ w =197. In terms of time unit u, the total time saved is 3849535 out of total time being 19680300. The spee up is 1.24.

Both the sequential and parallel algorithms are implemented in Fortran 77 and executed by Sparc Station. The results are about 10 seconds for a randomly generated m=200 matrix after w=197 sweeps. The parallel algorithm does not show the advantage is just due to a lot of the waiting time spent on distributing data, sending and receiving results between tasks.

In order to reduce the number of message passing we can combine module A and module B as one slave, and also combine module C and module D as another slave, then the number of message passing after w sweeps will be changed from 5w-2 to w. Then after w=197 sweeps, the actual parallel sweeping time will go down from 10 seconds to 8.4 seconds. And the speed up can be reached to 1.19 in practice.

6. Conclusion

In this study we gain a lot of experience of doing parallel computation by using a networked workstation clusters even though we don't have any access to a parallel computer. We found that it is very easy to develop and implement a parallel algorithm under PVM although the debugging is difficult. However, the disadvantage of using this distributed memory is that more waiting time will be spent on message passing. The reason why the speed up for this parallel algorithm can only reach to a maximum of 1.33 is due to fine grain size and unbalanced load. In addition, in order for the parallel algorithm to be advantageous over a sequential algorithm, a lot of sweeping must be done in the computation. This will cause the round-off error being significant. We will look for other parallel algorithms for computation of analyses of regression and experimental designs. A more user friendly extension of PVM called Heterogeneous Network Computing Environment(HeNCE) can be used([5]).

References

[1] Geist A., Beguelin A., Dongarra J., Jiang W., Manchek R., Sunderam V. (1993). PVM3 User's Guide and Reference Manual. Oak Ridge National Laboratory.

[2] Goodnight J.H. (1978). The Sweep Operators: Its Importance In Statistical Computing. Interface Foundation.

[3] Heiberger R.(1989). Computation for the Analysis of Designed Experiments. John Wiley and Sons.

[4] Dongarra J., Geist A., Manchek R., Sunderam V. (1993). Integrated PVM Framework Supports Heterogeneous Network Computing. Oak Ridge National Laboratory.

[5] Dongarra J., Geist A., Manchek R., Sunderam V. (1994). The PVM Concurrent Computing System: Evolution, Experience and Trends. Oak Ridge National Laboratory.

[6] Kennedy W.J., Gentle J.E. (1980). Statistical Computing. M. Dekker Inc.

[7] Douglas C., Mattson T., Schultz M. (1993). Parallel Programming Systems for Workstation Clusters. Computer Research Report, Yale University.

[8] Quinn M. (1994). Parallel Computing: Theory and Practice. 2nd edition. McGraw-Hill Inc.

Graphically Analyzing Computer Log Files

Stephen G. Eick and Paul J. Lucas

AT&T Bell Laboratories
Room IHC–1G–351
1000 East Warrenville Road
Naperville, IL 60566
eick@research.att.com

Keywords: software visualization, dynamic graphics, log files, Unix commands

SUMMARY

Computers generate log files containing reports on system performance, status, and faults. To analyze these log files more efficiently, we have developed an interactive visualization system, *SeeLog*,™ that displays temporal patterns and facilitates exploratory analysis of large log files. We apply our system and visualization techniques to analyze command accounting log files from a Unix compute server, although our motivating example was log files generated by software development lab testing.

1. Introduction

Many computer systems generate log files as part of their normal operation. Such files typically contain reports on system performance, status, and software faults. The reports are often free-format and time-stamped. These files are used by engineers for detecting and correcting system problems, hopefully before they become service-affecting. One attribute common to many log files is that they often contain many unimportant reports. These "noise" reports can clutter log files, obscure important reports, and thereby result in real problems going undetected.

Although our motivating example comes from analyzing log files created during the software development process, our analysis technique applies to other log files equally well. To illustrate our technique, we use the Unix System V command accounting facility. This log file contains a report for each command executed and is automatically generated as part of standard operations. It contains a detailed history of the machine's activity and it is used by system administrators for performance tuning, security monitoring, and could be used for usage billing. We find it particularly interesting because, by studying the logs from one of our own machines, we gain insight into how we in a research department use computing resources. By analyzing this data, we have gained some interesting insights into our own work patterns.

2. Visualization Technique

Our log file analysis paradigm involves two steps: parsing and visualization. Parsing a complicated log file involves lexicographically scanning it to note the times, types, and locations of all reports. This step can be done using tools like `grep`, AWK,[1] Perl,[2] or even a C program. The data from the scanning are placed in a table that is the input to *SeeLog*.

To create a visual display of a log file, the reports are arranged chronologically and grouped by type. Each report is then represented as an angled "tick mark" on a grid with time running along the x-axis and report type along the y-axis. The report types listed along the y-axis may be placed into bands of related types. The result is a pattern of horizontal bands, each containing a number related of rows, with ticks indicating occurrences. (See Figure 1.)

Within each band, there are rows for the distinct values of each type. The type name is printed at the left side of the display and the type value is printed next to its corresponding row. The rows may be sorted in decreasing tick mark frequency or in alphabetical order.

In most datasets, there are several dimensions of type information. For example, in the command accounting dataset, the type information includes the user-id, number of characters transferred, and process size. There are three methods that the tick marks encode type information: rows (primary method), color, and angle. The color and angle of each tick mark may encode different dimensions, but often redundantly encode the same dimension.

3. An Example

The display of the log file in Figure 2 shows ten hours of data from 4:00 to 13:59 from our department's compute server. During that period, 6179 command accounting reports were generated. The display shows two bands of reports: *system commands* (upper band)—those executed by either the `root`, `adm`, or various daemon logins and *user commands* (lower band)—those executed by ordinary users. The system commands are sorted alphabetically by command name; the user commands are sorted descendingly by the number of times each command was executed. The tick marks are display color- and angle-coded by the user-id. The user-ids are color-coded according to the interactive color scale on the left side of the display. The total number of occurrences for each command is shown on the right side of the display in the form of a bar chart, and the total number of occurrences for all commands is shown on the bottom of the display in the form of a stacked histogram. The slider in the bottom-left corner controls the bin size for the stacked histogram and is currently set at five minutes.

Many things are apparent from the display in Figure 2. The system commands `chkconfig` and `rpc.mount`, spawned by `sh` (Bourne shell), execute continuously throughout the ten-hour period. These involve the network file system (NFS). Another sequence of commands is executed hourly, on the hour, and involves accounting and periodic administrative tasks.

The user commands follow a different pattern. The stacked histogram at the bottom of the display shows that there was little user activity before 9, between 10 and 11 and during the noon hour. On this particular day, there was a department seminar between 10 and 11 and a lunch for our visitor. The most popular user command was the `CC` shell script, the front-end for the C++ compiler, which executed 1170 times.

There are large bunches of commands executed by user-id `pjl` (Paul Lucas). Those "waves" of commands were all started by the `CC` command. (He was actually compiling *SeeLog* a few times.) The first few were recompiles of selected object files; the compile performed around noon was a complete recompile. Some of the commands have tails indicating that they ran for a noticeable length of time. A few commands have tiny tails, particularly `pjl`'s `make`s and `CC`s, that are at different heights. The height of the tails varies when they would otherwise overlap on the display. A `make` command typically executes several other commands in sequence. On a single processor machine we would expect the commands spawned by the `make` to be executed one after each other with no overlap. The `make` command on our multi-processor compute server can make object files in parallel. The overlapping tails are instances when parallelization occurred, since the commands are executing concurrently.

The first set of user commands were executed by user-id `eick`. He started `ksh` (Korn shell) and read mail at 6:50 (from home) and executed several other commands at 7:47 (at work).

4. Summary

The *SeeLog* system embodies a graphical technique for visualizing large, computer-generated log files. The system graphically displays log file reports and provides interactive mechanisms for manipulating the display. All of the reports are displayed on a single grid as tick marks, using position, color, and angle to encode the type, time, attributes and subattributes of each report. Using our technique we have analyzed log files with over 80,000 error messages, in a fraction of the time required by conventional methods. This log file analysis technique generalizes to analyzing any stream of time-stamped, typed reports. This includes output from transactions systems, data networks and even electronic-mail logs.

REFERENCES

1. Alfred V. Aho, Brian W. Kernighan, and Peter J. Weinberger. *The AWK Programming Language*. Addison-Wesley, 1988.
2. Lawrence Wall and Randal L. Schwartz. *Programming perl*. O'Reilly & Associates, Inc., 1990.

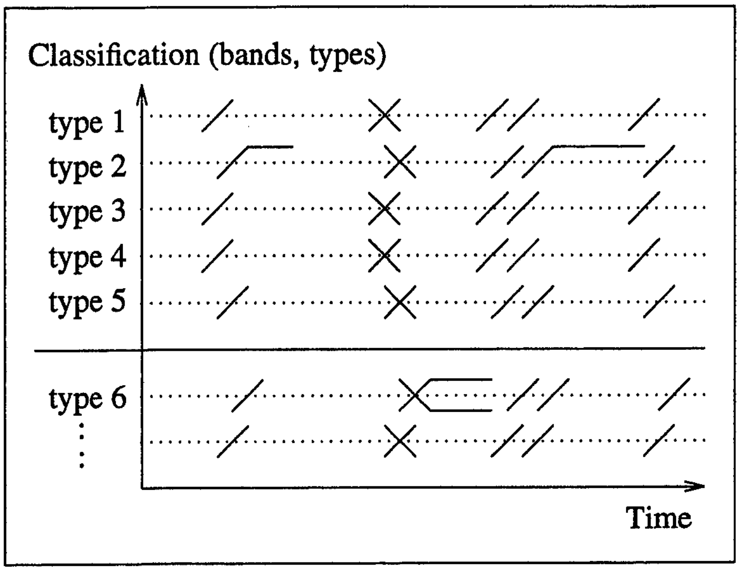


Figure 1. Log File Display

Each tick mark represents one report and is positioned on a grid chronologically and grouped by type. The *x*-axis encodes time and the *y*-axis type.

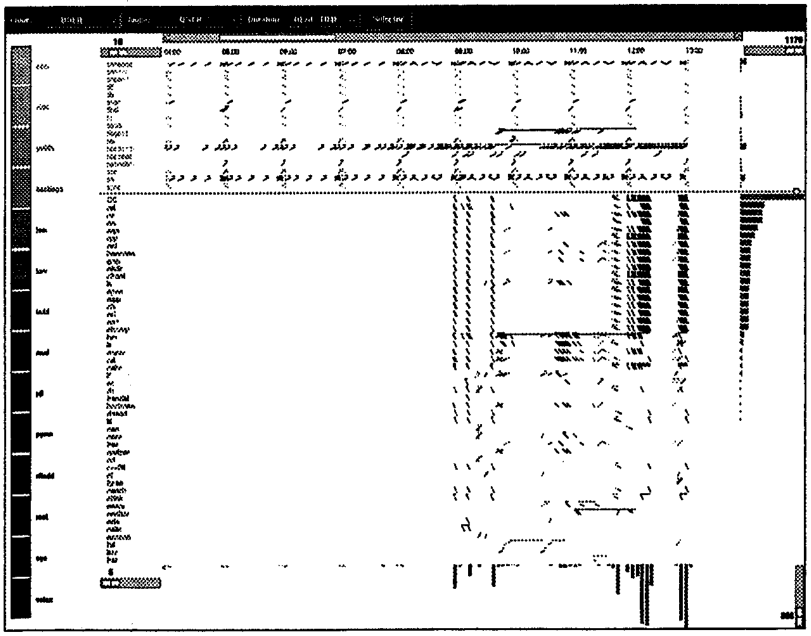


Figure 2. Who did what: Coded by user-id

Each tick mark represents one Unix command or shell script that executed during a ten hour period on our compute server. The tick marks are positioned on a chronological-by-type grid and color- and angle-coded to show the user-id executing that command.

The Multi-String Rearranging Memory and Its Use in Statistical Computing[1]

P. N. Armstrong
18 Elk Run
Monterey, CA 93921

and

R. R. Read
Department of Operations Research
Naval Postgraduate School
Monterey, CA 93943

Abstract

The Multi-String Rearranging Memory (MSRM) is a computer memory system designed for use with standard (e.g., IBM 486 or larger) computers. Simultaneous input, output, and data rearrangement operations are permitted when it is installed in a computing system. Such common computations as formation of the transpose of a matrix, order statistics ranking operations, construction of empirical cumulative distributions, quantiles, etc., require no more time than linear time.

Other operations for which the MSRM is designed include "skimming" and searching. When the MSRM is used for skimming, the largest (smallest) *m* of a list of *n* entries can be selected in the amount of time consumed by a single scan of the *n* entries. This use of the MSRM permits such operations as removal of outliers, the extraction of sets of extremal observations, and trimming of data. When the MSRM is used for searching, *m* entries may be matched with *n* entries in the MSRM; the amount of time required for this operation is the amount of time needed for transmission of the *m* entries to the MSRM. Delay between successive members of the list of *m* entries is not required.

Performance characteristics, statistical uses, and intrinsic cost of the MSRM are discussed in the paper.

0. Introduction.

The Multi-String Rearranging Memory (MSRM) is a computer memory system. It is hardware, not software. It consists of standard RAM and some control circuitry suitable for VLSI construction. The RAM can be used as ordinary RAM storage when it is not being used for the special purposes described below. The cost of the MSRM is dominated by the cost of its RAM.

The functional characteristics of the MSRM are described in Ref. 1. The principal operations and their performances will be reviewed in the first section. Discussion of the speed advantages of the MSRM appears in the second section. The third section describes some advantages of its use in statistical computing. In particular, a somewhat detailed example related to isotonic regression is presented. It serves to indicate an advantage of the usage.

1. Specialized Operations.

A main data management operation is that of sorting; i.e., the placement of data in increasing (decreasing) order. Most conventional computers use $n \log_2(n)$ serial operations to sort data where n is the number of records. The MSRM sorts records in linear time. More specifically, the n records are read in serially without any delay between successive records. When they are written out serially, again without any delay, they will be in sorted order. One can think of filling a pipe; during the input process the records undergo some rearrangement. The final rearrangement takes place during the output process.

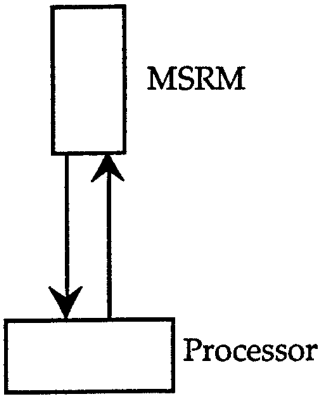
The insert operation can be performed, also in linear time. On occasion we have need to withdraw some data, modify it, and put in back in. This takes time proportional to the number of records that are modified.

A data skimming operation is valuable and fast. Suppose we desire the m largest (smallest) records. These can be placed in the MSRM in the time it takes for a single pass of the data. Thus this operation is just as fast as the seemingly similar operation of screening all records that satisfy a given inequality or property.

Basic searching operations can also be performed in linear time. If an ordered sequence of size n is placed in the MSRM and q individual query records are submitted, then these records can be input serially, without delay, and the records with matching keys can be located in time proportional to q.

[1] The text of this paper was prepared by R. R. Read. The portion of the paper that pertains to the MSRM was obtained in part from various papers prepared by myself. We suggest that any inquiry regarding the procedures mechanized with the MSRM be directed to me; it is unfortunate that proprietary considerations interfere with publication of some of the details of the MSRM. Philip N. Armstrong

There is an additional advantage in that input and output operations can take place simultaneously; as one record is going in, another one can be withdrawn. If this is done in sort mode, then the input records must be for a new sorted file. If non-destructive output from the MSRM is desired, the records received by the processor may be reinserted into the MSRM either as a new file or into the original file. The bi-directional transmission is indicated in Figure 1.



General Computing System with MSRM

Figure 1

The use of the MSRM relieves the computer's system control of many of its tasks. One requires but a single address in the MSRM and writing the file, all records serially, to that address. On output, the records are read serially from that address. Thus the MSRM behaves as a pipe; it has the added advantage that the records are rearranged while being placed in and withdrawn from "the pipe".

2. Timing Comparisons.

The importance of sorting has received some recent attention [4, 5]. Typically computer centers use software systems to perform sorting operations when sorting is required. Normally these calls are not visible to the user. Time spent sorting is buried in the elapsed time of a job and the number of calls to sorting operations is also lost.

Some idea of the size and speed of an MSRM system may be gained by comparing it with a large computer, e.g., an IBM 9012. The MSRM, operating at currently feasible frequency, is faster than the IBM system, according to the published IBM specifications [4]. An MSRM system can be constructed in accordance with the parameters:

Capacity: 1.2 gigabytes; MSRM word size: 4 bytes
Record Length is any fixed number of words
Memory Input/Output speed: 10^7 words per second

The amounts of time required for sorting files of various sizes with the IBM system and with the MSRM are tabulated in Table 1. In it, the column I/M is the ratio defined by the amount of time consumed by the RAM (IBM) system divided by the amount of time consumed by the MSRM system.

The file and record sizes used in Table 1 may seem larger than those contemplated in many statistical computations. Generally, the sorting advantage is a factor of $\log_2(n)$. The MSRM advantage increases with the number of records in a file (data set); and never is it at a disadvantage.

Table 1
MSRM Sort Timing

File Size (Megabytes)	MSRM Elapsed[2] M Time	IBM Elapsed[3] I Time	Ratio: I/M
10	.25	8	32
20	.5	13	26
40	1.0	26	26
80	2.0	50	25
150	3.75	93	24.8
300	7.5	182	24.2
600	15	366	24.4
1200	30	725	24.2

3. Statistical Computing.

The sorting and skimming capabilities of the MSRM serve nicely for the elementary operations in data analysis. Suppose the data are $(x_1, x_2, ..., x_n)$ and we require the order statistics, the empirical distributive function, histograms, and ranks. The order statistics

[2] The quoted amounts of time do not include time for access to the mass store, if any, in which the data is stored; it is assumed that the file passes to the MSRM at the rate of 10^7 bytes per second. It is also assumed that, since data can pass from the MSRM in sorted order, that it is not necessary to record the file in mass memory after it is received in the MSRM. The computed time is thus, for the file of 300×10^6 bytes, $300 \times 10^6 / 4 \times 10^7 = 7.5$ seconds. This would also be the output time if output is required before other uses for the sorted file requires such storage.

[3] The IBM data is published in Ref. 4. In trials to determine the accuracy of the data assumed here, an Amdahl computer was used (the Amdahl 5995-700A installed at the Naval Postgraduate School at Monterey, CA) with the collaboration of the Defense Manpower Data Center at Monterey. The Amdahl system is somewhat slower than the IBM system, but still the input/output time was reported to be negligible compared to the sort time. This suggests that the timing shown for the MSRM is at least nearly attainable in the large IBM or Amdahl systems and neglect of the input/output time is not a distortion of the MSRM performance.

$$x_{(1)} \leq x_{(2)} \leq \ldots \leq x_{(n)}$$

can be obtained in linear time. The pairs $(x_{(j)}, 1/n)$ effectively form the empirical cdf. The construction of histograms follows directly from the order statistics for the empirical cdf. Indeed the latter is useful for the construction of histograms having equi-probable cells.

The generation of ranks seems to require an extra step: form the pairs (x_j, j) as records and rearrange the file according to increasing x_j in the MSRM. The output file has records of the form $(x_{(i)}, j_{(i)}, i)$ and $j_{(1)}, \ldots, j_{(n)}$ will be the inverse permutation of the ranks. The indices i are appended during the output process. Then the pairs $(j_{(i)}, i)$ are input to the MSRM and sorted according to increasing values of the $\{j_{(i)}\}$. The output will be the pairs (j, r_j) where $\{r_j\}$ is the set of ranks.

The skim operation allows immediate trimming of the data set without prior ordering. The top $m = <pn>$ of the data (where $0 < p < 1$) can be removed with a single pass. That top portion will reside in the MSRM and can be withdrawn in sorted order. Such operations permit rapid access for the study of extrema and outliers. The remainder of the data will be in its original order

Both tails can be trimmed virtually simultaneously: the observations enter a single address of the MSRM which will accept (and order) the first m. As the next observation goes in it is compared with the smallest of the first m; the larger of these two replaces the smaller which in turn is sent to a different MSRM file address (for skimming the lower $<qn>$ observations in a similar manner). This continues serially and the file retained at the original address will contain the $<pn>$ largest while the file retained at the second address will contain the $<qn>$ smallest. These latter will be in decreasing order. Of course $p + q < 1$. The remainder of the data returns to ordinary storage (or perhaps is discarded).

The summation of the observations in a large data set may be accomplished profitably using the MSRM in some instances. If we insert $(x_1, \ldots, x_n)$ for withdrawal in decreasing order of magnitude, then a sharp approximation to the total may be computed without summing them all. More specifically we can accumulate

$$S_m = x_{(1)} + x_{(2)} + \ \ldots + x_{(m)}$$

and the error in S_m is dominated by

$$|n-m| \cdot |x_{(m+1)}|.$$

We close with a more complex example of exploitation of the MSRM in statistical computing. It serves to illustrate the effective use of the MSRM in a statistical estimation method. Consider isotonic regression in the simply ordered case. We use the notation of [2, 3]. We hope to convince the reader that the pool-adjacent-violators-algorithm (PAVA) can be modified to exploit the capabilities of the MSRM and that the estimates can be produced in less time.

The setting is a set of k linearly ordered entities

$$x_1 < x_2 < x_3 < \ldots < x_k$$

and to each there is a value, $g_i = g(x_i)$ and a weight $w_i > 0$. The goal is to compute the isotonic regression values $\{g_i^*\}$ for $i = 1, \ldots, k$. These are the ordered values

(1) $$g_1^* \leq g_2^* \leq \ldots \leq g_k^*$$

that most closely resemble the original $\{g_i\}$. The sense for which this holds is described in the first chapters of [2, 3]. The development considers the cumulative sum diagram (CSD) defined by the scatter plot of $\{W_j, G_j\}$ for $j = 0, 1, \ldots, k$ where $W_0 = G_0 = 0$ and

$$W_j = \sum_{i=1}^{j} w_i \quad \text{and} \quad G_j = \sum_{i=1}^{j} w_i g_i.$$

See Figure 2 for an example. The development of the PAVA involves the construction of the greatest convex minorant (GCM) of the CSD; i.e., the supremum of all convex functions whose graphs are below the CSD. It is known that g_i^* is the left derivative of the GCM at W_i for $i = 1, \ldots, k$.

The first step is to check whether

$$g_1 \leq g_2 \leq \ldots \leq g_k$$

is the original condition. If so then $g_i^* = g_i$ for all $i = 1, \ldots, k$ and there is nothing more to be done. The MSRM can make this check in linear time. (If this condition is satisfied then the CSD is a scatter plot which, when the successive points are connected with straight line segments, forms a convex function.)

If the first check fails, then the PAVA seeks a set of levels or blocks of subscripts so that the $\{g_i^*\}$ are constant within blocks but variable from block to block. Initially each subscript is a block. The polling of subscripts to form larger blocks is accomplished through the selection of "violators", i.e., adjacent pairs having the property $g_i > g_{i+1}$. This pair combines two blocks into one using the weight of average

$$(w_i \, g_i + w_{i+1} \, g_{i+1}) \, / \, (w_i + w_{i+1})$$

for its value and $(w_i + w_{i+1})$ for its weight. Then the monotone inequality (1) is checked again. The algorithm is finished if it is satisfied; otherwise choose another violator pair and repeat the method.

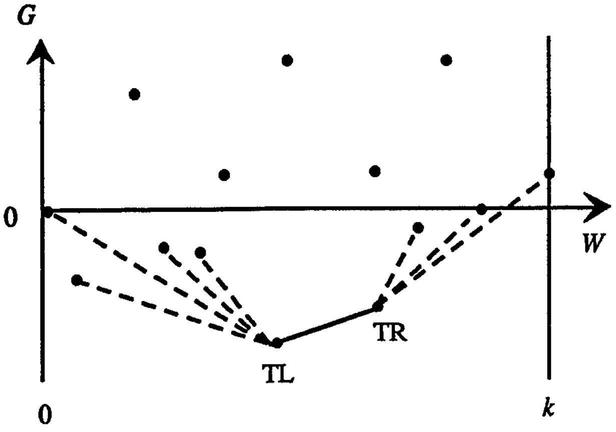
The construction of blocks is done sequentially. That is, a block size grows in increments of one during each iteration. It is possible to speed up this process.

In the interest of brevity let us suppose there are many violators. Our goal is to illustrate the usefulness of the MSRM without unnecessary details.

Form records of the type

$$\{W_j, G_j, j\} \text{ for } j = 0, 1, \ldots, k$$

and send this file to an address in the MSRM so that the records can be withdrawn in increasing order of the $\{G_j\}$. It is convenient to include the index j in each record. When withdrawing records from the MSRM we must make comparisons and assign each to either the left pool, LP, or the right pool, RP, according to whether $j < \text{TL}$ or $j > \text{TR}$. We also need a base value for computing slopes: GL for the left pool and GR for the right pool. Initially GL = GR = $\min\{G_j\}$ and TL = TR = j of the first record to come out of the MSRM. The common horizontal accumulated weight is WL = WR = W.



Scatter Plot that Supports the Cumulative Sum Diagram Steps to Construct the Greatest Common Minorant

Figure 2

The diagram can be used to visualize an iterative step in the construction of the GCM. The records come out in increasing order of $\{G_j\}$. The record is assigned to LP if $j < \text{TL}$ and to RP if $j > \text{TR}$. As the records come out we compute the appropriate slope

$$\frac{\text{GL}-\text{G}}{\text{WL}-\text{W}} \quad \text{or} \quad \frac{\text{G}-\text{GR}}{\text{W}-\text{WR}}$$

and send records (SL, W, j) to a new file in the MSRM for the left pool, and records (SR, W, j) to another new file for the right pool. The left pool is sorted in descending order, (i.e., ascending order of magnitude), and the right pool is sorted in ascending order. The iterative step stops when either $j = 0$ or $j = k$.

For definiteness suppose we are stopped at $j = 0$. Then we address the file that contains the SLs (i.e., the left slopes) and extract the first one (the largest slope). We set $g_i^* = \text{SL}$ for all $j < i \leq \text{TL}$; we update TL = j, WL = W and purge the left slope file. Next check the updated (1) for violators at TL or to its left. The left pool can be abandoned if there are none. We also return the $\{W, G, j\}$ records, for $j \leq \text{TL}$, to the original MSRM file address. The others are discarded. Then start another iteration. It may begin a new left pool or it may complete the existing right pool or both. Note that it is never necessary to use slopes of segments connected to CSD points that are above zero for the left pool, or above G_k for the right pool. It is clear that the process will finish and produce the desired isotonic regression.

The algorithm, with obvious modification, can be used to find the convex hull of a two dimensional point cloud. Also there is an obvious simplification to this version of the PAVA. One can add a known constant to all of the $\{g_i\}$ so that there are no negative values. This will circumvent the need for a left pool.

The speed of the procedure rests on the fact that there is but a minimal amount of addressing. Each address merely opens a "pipe" from which all needed information appears and in the proper order. The number of addresses is not determined by the magnitude of the data set; it appears to be small, perhaps 3 or 4. Also, the original PAVA appears to have more intermediate steps, more overt comparisons, and more weighted averages to compute. It would be interesting to have timed comparisons for a variety of cases.

The advantages of using an MSRM lie in simplified programming, fewer address and fetch operations, and greater speed.

References

[1] P.N. Armstrong (1993). *Data Rearrangement and Real-Time Computation*, RAND, ISBN 0-8330-1340-8.

[2] T. Robertson, F.T. Wright, R.L. Dykstra (1988). *Order Restricted Statistical Inference*, Wiley.

[3] R.E. Barlow, D.J. Bartholomew, J.M. Bremner, H.D. Brunk (1972). *Statistical Inference under Order Restrictions*, Wiley.

[4] IBM (1993). DFSORT Tuning Guide Release 12, SC26-3111-00.

[5] D.H. Bailey, E. Barszcz, L. Dagum, H.D. Simon (1994). RNR Technical Report, RNR-94-006, NASA, Ames Research Center, Moffett Field, CA, 94035.

Fast Multidimensional Density Estimation based on Random-width Bins

Leonard B. Hearne[1] and Edward J. Wegman[2]
Center for Computational Statistics
George Mason University
Fairfax, VA 22030

Abstract

Histogram-type density estimators have some notable computational advantages over other forms of density estimation by virtue of the WARPing algorithm. However, traditional fixed-bin-width have less than satisfactory smoothing properties, being too coarse in regions of high density and too fine in regions of low density. Scott (1992) suggests the ASH algorithm as a means of overcoming these problems, but the ASH algorithm is computationally intensive somewhat negating the benefits of WARPing. Wegman (1975) proposed a variable bin-width technique for one dimensional density estimators and used sieve-type methods to show strong consistency results that did not depend on smoothness properties of the underlying density. In this paper, we extend this idea to high-dimensional, variable bin-width meshes. The boundaries of the bins are determined by a random subsampling of the observations. An extension of the WARPing algorithm may still be used for fast computation. We give combinatorial arguments for calculating the number of bins and also the conditional expectation and variance of the number of observations per bin. Conditional on the random hyper-rectangular tessellation, we calculate the maximum likelihood density estimator.

Introduction

In this paper, a density estimation method is developed that is computationally more tractable than kernel density methods, and has better smoothing properties than traditional fixed binning methods. The basic method is easy to describe in one dimension. Randomly select a subset of m observations $\left\{Y^*\right\}$ from a set of n observations $\{Y\}$, $m < n$, together with the $max\{Y\}$ and $min\{Y\}$. Order the set $\left\{Y^*\right\}$ in the set $\left\{Y^*_{(\cdot)}\right\}$. A set of random width bins $\{B\}$ can be can be constructed using adjacent elements in the set $\left\{Y^*_{(\cdot)}\right\}$. Then attribute the probability mass of all observations in $\{Y\}$ to the bins in $\{B\}$. The probability density on an element $B_i \in \{B\}$ is the relative probability mass on B_i divided by the length of B_i, *cf*. Wegman (1975) and Hearne and Wegman (1991). There are many ways to generalize these results to a d-dimensional support space. The generalization that we have adopted here is to define random-width d-dimensional rectangular bins generated by a random sample from the set of observations.

Random-width d-Dimensional Bin Tessellation

Given a set of n observations, $\{Y\}$, in a d-dimensional Euclidian space, let A^d_n be the minimum d-dimensional rectangular cover of $\{Y\}$. Each observation $Y_j \in \{Y\}$ can be written in the form

[1] The research of Dr. Hearne was supported by the National Security Agency under contract MDA 904-92-H-3082

[2] The research of Dr. Wegman was supported by the Army Research Office under Contract DAAL03-91-G-0039, the Office of Naval Research under Grant N00014-92-J-1303, and the National Science Foundation under Grant DMS9002237.

$Y_j = \left(Y_j^1, Y_j^2, \cdots, Y_j^d\right)$. Then A_n^d can be defined by the set of maximum and minimum values for the d coordinate axes,

$$A_n^d \equiv \left\{x \in \Re^d \colon x^i \geq min(Y^i) \wedge x^i \leq max(Y^i)\right\}.$$

A d-dimensional rectangular tessellation of A_n^d can be generated by selecting a random subsample of N observations $\{S_N\}$ from $\{Y\}$. For each of the d coordinate axes let $\left\{S_N^i\right\}$ be the set of the i^{th} coordinate for all $Y \in \{S_N\}$ together with $max(Y^i)$ and $min(Y^i)$. Let $\left\{S_{(\cdot)}^i\right\}$ be the ordered set of unique elements in $\left\{S_N^i\right\}$ and $s^i = card\left\{S_{(\cdot)}^i\right\}$. A set of one dimensional bins, $\{B^i\}$, can be generated for each of the d coordinate axes by adjacent elements in the set $\left\{S_{(\cdot)}^i\right\}$, and $card\{B^i\} = s^i - 1$. The d-dimensional rectangular random tessellation $\left\{B_N^d\right\}$ of A_n^d can then be generated by the cross product of the sets of one dimensional bins for each coordinate axis;

$$\left\{B_n^d\right\} = \{B^1\} \times \{B^2\} \times \cdots \times \{B^d\}, \text{ and}$$

$$m = card\left\{B_n^d\right\} = \prod_{i=1}^{d} (s^i - 1).$$

The upper bound on the cardinality of the set of one dimensional bins that are generated for each of the coordinate axes is $s^i - 1 \leq N + 1$, $1 \leq i \leq d$, since the random sample $\{S_N\}$ may have observations that contain $max(Y^i)$ or $min(Y^i)$, observations are recorded only to finite precision, and computers operate on a subset to the rational numbers. The cardinality of the tessellation $\left\{B_N^d\right\}$ then has an upper bound, given the random subsample $\{S_N\}$ of

$$m = card\left\{B_n^d\right\} = \prod_{i=1}^{d} (s^i - 1) \leq (N+1)^d.$$

In Figure 1 a set of observations $\{Y\}$ in $\Re^2$ have values $max(Y^1)$, $min(Y^1)$, $max(Y^2)$, and $min(Y^2)$. These values define the minimum 2-dimensional rectangular cover A_n^2 of $\{Y\}$. A random subsample of observations is drawn from $\{Y\}$, $\{S_3\} \equiv (p_1, p_2, p_3)$. These three points together with the maximum and minimum values for each of the coordinate axes generate the set of bins $\{B_n^2\}$ of A_n^2.

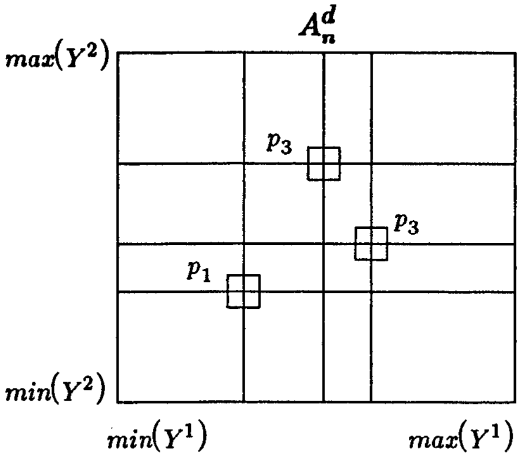


Figure 1

The tessellation $\left\{B_n^d\right\}$ of A_n^d is adaptive in the sense that the elements of the tessellation tend to be large where the observations are sparse and small where the observations are not sparse.

Conditional Expectation and Variance of the Number of Observations per Bin

Let B_k, $1 \leq k \leq m$, be the k^{th} d-dimensional bin in the tessellation $\left\{B_n^d\right\}$ of A_n^d, and let Z_k be the number of observations in $\{Y\}$ that are in B_k. The expected value of Z_k given the tessellation $\left\{B_n^d\right\}$ is the number of observations that might be attributed to the k^{th} bin times the probability that the d-dimensional random variable X is in the k^{th} bin;

$$E\left[Z_k \mid \left\{B_n^d\right\}\right] = (n - N)P(X \in B_k).$$

Let U_j^i, $1 \leq i \leq d$, be the empirical probability mass on the j^{th} one dimensional bin, $1 \leq j \leq s^i - 1$, for the i^{th} coordinate axis,

$$U^i_j = F\left(Y^i_{(j-1)}\right) - F\left(Y^i_{(j)}\right) = P\left(X^i \in B^i_j \mid \{B^i\}\right).$$

Using order statistical arguments, *cf.* Rohatgi (1976) pp.575-580, it can be shown that;

$$E\left[U^i_j \mid \{B^i\}\right] = \frac{1}{s^i - 1},\ 1 \leq j \leq s^i - 1, \text{ and}$$

$$V\left[U^i_j \mid \{B^i\}\right] = \frac{s^i - 2}{(s^i - 1)^2 s^i}.$$

Since the tessellation $\left\{B^d_n\right\}$ of A^d_n is generated by the cross product of the one dimensional bins on each of the d coordinate axes then the probability mass that is on a given d-dimensional bin $B_k \in \left\{B^d_n\right\}$, given the tessellation $\left\{B^d_n\right\}$, is;

$$E\left[U_k \mid \left\{B^d_n\right\}\right] = \prod_{i=1}^{d} \frac{1}{s^i - 1},\ 1 \leq k \leq m, \text{ and}$$

$$V\left[U_k \mid \left\{B^d_n\right\}\right] = \prod_{i=1}^{d} \frac{s^i - 2}{(s^i - 1)^2 s^i}.$$

Multiplying by the number of observations that might be attributed to a d-dimensional rectangular bin, $n - N$, and applying the inequality bounding the cardinality of the number of bins in the tessellation;

$$E\left[Z_k \mid \left\{B^d_n\right\}\right] \geq \frac{n - N}{(N+1)^d},\ 1 \leq k \leq m, \text{ and}$$

$$V\left[Z_k \mid \left\{B^d_n\right\}\right] \geq \frac{(n - N)^2 (N-1)^d}{N^{2d}(N+1)^d}.$$

A Class of Probability Density Estimators

Let n be the number of observations in the set of observations $\{Y\}$, and let n_k be the number of observations in the k^{th} rectangular bin in the tessellation $\left\{B^d_n\right\}$. Let $W(N_k)$ be the probabilistic mass of observations in the tessellation generating set $\{S_N\}$ that are attributed to an adjacent bin in the tessellation $B_k \in \left\{B^d_n\right\}$ by the function $W(\cdot)$. And let C_k be the d-dimensional content of the k^{th} element of the tessellation. Then we can define a class of probability density estimators on a tessellation $\left\{B^d_n\right\}$ by;

$$\hat{f}(x \in B_k) = \frac{n_k + W(N_k)}{n \cdot C_k} \text{ and}$$

$$\hat{f}\left(x \notin \left\{B^d_n\right\}\right) = 0.$$

This class of probability density estimators is constant on each bin in the tessellation, and the content of each of the d-dimensional bins in the tessellation C_k is easily computed. The probabilistic mass attribution function $W(\cdot)$ is closely related to the likelihood function.

The Likelihood Function

The likelihood function was introduced as a means for optimizing the parameter values in the parametric density estimation setting so that the fitted parametric function would best fit a set of observations. In the nonparametric setting the likelihood function has utility if there is a variable in the class of density estimators. The weight that is attributed to bins in the tessellation by observations in $\{S_N\}$ is variable and can be used to optimize the likelihood function.

The likelihood function for this class of probability density estimators is

$$L(x) = \prod_{j=1}^{n} \frac{n_k + W(N_k)}{n \cdot C_k},$$

the product of the density estimates for each of the observations. But the class of density estimators that are presented here are estimators on the set of bins in the tessellation of A^d_n so the likelihood function can be reformulated in terms of the elements of the tessellation;

$$L(x) = \prod_{k=1}^{m} \left(\frac{n_k + W(N_k)}{n \cdot C_k}\right)^{(n_k + W(N_k))}.$$

Taking the first derivative of the log of the likelihood function with respect to $W(N_k)$;

$$\frac{d}{dW(N_k)} log L(x) = \sum_{k=1}^{m} \left(\frac{n_k}{n_k + W(N_k)} + \frac{W(N_k)}{n_k + W(N_k)} \right) + \sum_{k=1}^{m} \left(log(n_k + W(N_k)) - log(n \cdot C_k) \right).$$

If the first derivative is set equal to zero and solved for $W(N_k)$ then the estimator will be optimized, either maximized or minimized depending on the sign of the second derivative of the log of the likelihood function. Taking the second derivative of the log of the likelihood function;

$$\frac{d^2}{dW(N_k)^2} log L(x) = \sum_{k=1}^{m} \frac{n_k}{n_k + W(N_k)}.$$

The second derivative of the log of the likelihood function with respect to $W(N_k)$ is positive on all bins in the tessellation that have observations in them, $n_k > 0$, and is undefined where $n_k = 0$. The likelihood function is thus convex and the likelihood function is maximized when the probabilistic mass of all observations in $\{S_N\}$ are attributed to the adjacent bin where $\frac{n_k + 1}{C_k}$ will be largest.

A Random Bin-width Warping Algorithm

For the proposed probability density estimation method to be of utility it is important that density estimates be readily computable, given a set of n observations, $\{Y\}$, in a d-dimensional Euclidian space. The principal computational complexity is in the attribution of observations to bins in the tessellation, $\left\{B_n^d\right\}$, of the minimum d-dimensional rectangular cover of $\{Y\}$, A_n^d. In conventional fixed width binning methods an algorithm called warping has been developed that increases the speed and reduces the computational complexity for attributing observations to bins in the tessellation. This algorithm has been extended to variable bin-width tessellations.

Given N the number of observations in the random sample of observations used to generate the rectangular bins in the tessellation, the cardinality of the set of bins, m, is bounded by;

$$m = card\left\{B_n^d\right\} = \prod_{i=1}^{d} (s^i - 1) \le (N+1)^d.$$

For each coordinate axis there is an upper bound on the number of one dimensional bins that can be generated. Let Bound_Values[i, j] be a matrix with the i^{th} row, $0 \le i < d$, corresponding to $\left\{S^i_{(\cdot)}\right\}$ and Bound_Value[i,0]=$min(Y^i)$. Then for each row i, $0 \le j \le s^i - 1$. Let Bin_Index[i, k] be a matrix with the i^{th} row a vector of integer indices into the matrix Bound_Values[i, j], with $0 \le k < w^i$, where w^i is the selected number of warping indices for the i^{th} coordinate axis, $s^i - 1 \le w^i$.

Let $b^i = min(Y^i)$ and $a^i = \frac{max(Y^i) - min(Y^i)}{w^i}$ for the i^{th} coordinate axis, $0 \le i < d$. For any point $x^i \in \left[min(Y^i), max(Y^i)\right]$ then the value

$$\text{Index} = \text{Truncate}\left[\frac{(x^i - b^i)}{a^i}\right]$$

is an integer in the range $0 \le \text{Index} < w^i$. Let the i^{th} coordinate axis and the k^{th} entry in the matrix Bin_Index[i, k] be the smallest index j into the matrix Bound_Values[i, j] such that

$$a^i(\text{Index} + b^i) \le \text{Bound\_Values}[i, j].$$

Then an efficient algorithm to compute the bin index for the i^{th} coordinate axis, $0 \le i < d$, is shown in the following code fragment.

```
Get_Bin_Index($i, x^i$)
    Table_Index = Truncate$\left((x^i - b^i)/a^i\right)$
    Index = Bin_Index[$i$, Table_Index]
    While($x^i >$ Bound_Values[$i$, Index]) Index++
    Return Index
```

The size of the number of warping indices, w^i, is specified by the user of the density estimation method. The question of how large w^i should be is of interest. We want to maximize the probability of selecting the correct bin index on the first attempt for each of the d coordinate axes. The bounds on the probability of selecting the correct bin index on the first attempt is;

$$P\left(x^i < \text{Bound\_Values}[i,\text{Bin\_Index}[i,\text{Table\_Index}]]\right) \geq \frac{w^i - s^i + 1}{w^i}.$$

The larger w^i is relative to $s^i - 1$, the larger the probability that the correct bin index will be computed on the first attempt. If the density function is symmetric then the expected value of the probability is $\frac{w^i - (s^i - 1)/2}{w^i}$.

Conclusions and Extensions

Random-width binning methods are a computationally tractable alternative to fixed-width binning methods. The size of the bins in a d-dimensional space are adaptive so that the bins will tend to be large where the observations are sparse and small where the observations are not sparse. Bounds on the expected value and variance of the number of observations that are attributed to each bin can be calculated, given the size of the subsample that is randomly selected from the set of observations to generate the d-dimensional bins. The likelihood function is convex a function that can be maximized or minimize to give a maximum entropy estimate by selecting the appropriate probabilistic weight distribution function $W(\cdot)$, *cf.* Hearne and Wegman (1992). By applying an extension to the WARPing algorithm, the computational complexity of the random-width binning method is only slightly more computationally intensive than fixed-width binning methods.

One of the natural extensions to random-width binning methods is to apply a resampling scheme, *cf.* Billard and LaPage (1992). Given smoothness assumptions about the underlying probability density, then the size of the set of observations, the dimension of the observations space, and the expected value and variance bound on the number of observations that are attributed to each bin might be used to find the optimal subsample size, and the number of resampling repetitions necessary to achieve the desired density estimate smoothness. Resampling in an optimal way is believed to be less computationally intensive than either kernel or ASH methods, *cf.* Scott (1992).

Bibliography

Hearne, L.B. and Wegman E.J. (1991). "Adaptive Probability Density Estimation in Lower Dimensions using Random Tessellations", *Computing Science and Statistics*, Keramidas, E.M. (ed.), 23 241-245, Interface Foundation of North America, Fairfax Station, VA.

Hearne, L.B. and Wegman E.J. (1992). "Maximum Entropy Density Estimation using Random Tessellations", *Computing Science and Statistics*, Newton J. (ed.), 24 483-487, Interface Foundation of North America, Fairfax Station, VA.

LePage, R. and Billard, L. (1992). *Exploring the Limits of Bootstrap.* John Wiley & Sons, New York.

Rohatgi, V.K. (1976). *An Introduction to Probability Theory and Mathematical Statistics.* John Wiley & Sons, New York.

Wegman, E.J. (1975). "Maximum Likelihood Estimation of a Probability Density Function" *Sankhyā Ser. A* **37** 211-224.

Global Tree Optimization: A Non-greedy Decision Tree Algorithm

Kristin P. Bennett
Department of Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180

Abstract

A non-greedy approach for constructing globally optimal multivariate decision trees with fixed structure is proposed. Previous greedy tree construction algorithms are locally optimal in that they optimize some splitting criterion at each decision node, typically one node at a time. In contrast, global tree optimization explicitly considers all decisions in the tree concurrently. An iterative linear programming algorithm is used to minimize the classification error of the entire tree. Global tree optimization can be used both to construct decision trees initially and to update existing decision trees. Encouraging computational experience is reported.

1 Introduction

Global Tree Optimization (GTO) is a new approach for constructing decision trees that classify two or more sets of n-dimensional points. The essential difference between this work and prior decision tree algorithms (e.g. CART [5] and ID3 [10]) is that GTO is non-greedy. For greedy algorithms, the "best" decision at each node is found by optimizing some splitting criterion. This process is started at the root and repeated recursively until all or almost all of the points are correctly classified. When the sets to be classified are disjoint, almost any greedy decision tree algorithm can construct a tree consistent with all the points, given a sufficient number of decision nodes. However, these trees may not generalize well (i.e., correctly classify future not-previously-seen points) due to over-fitting or over-parameterizing the problem. In practice decision nodes are pruned from the tree. Typically, the pruning process does not allow the remaining decision nodes to be adjusted, thus the tree may still be overparameterized. The strength of the greedy algorithm is that by growing the tree and pruning it, the greedy algorithm determines the structure of the tree, the class at each of the leaves, and the decision at each non-leaf node. The limitations of greedy approaches are that locally "good" decisions may result in a bad overall tree and existing trees are difficult to update and modify.

GTO overcomes these limitations by treating the decision tree as a function and optimizing the classification error of the entire tree. The function is similar to the one proposed for MARS [8], however MARS is still a greedy algorithm. Greedy algorithms optimize one node at a time and then fix the resulting decisions. GTO starts from an existing tree. The structure of the starting tree (i.e. the number of decisions, the depth of the tree, and the classification of the leaves) determines the classification error function. GTO minimizes the classification error by changing all the decisions concurrently while keeping the underlying structure of the tree fixed. The advantages of this approach over greedy methods are that fixing the structure helps prevent overfitting or overparameterizing the problem, locally bad but globally good decisions can be made, existing trees can be re-optimized with additional data, and domain knowledge can be more readily applied. Since GTO requires the structure of the tree as input, it complements (not replaces) existing greedy decision tree methods. By complementing greedy algorithms, GTO offers the promise of making decision trees a more powerful, flexible, accurate, and widely accepted paradigm.

Minimizing the global error of a decision tree with fixed structure is a non-convex optimization problem. The problem of constructing a decision tree with a fixed number of decisions to correctly classify two or more sets is a special case of the NP-complete polyhedral separability problem [9]. Consider this seemingly simple but NP-complete problem [9]: Can a tree with just two decision nodes correctly classify two disjoint point sets? In [4], this problem was formulated as a bilinear program. We now extend this work to general decision trees, resulting in a multilinear program that can be solved using the Frank-Wolfe algorithm proposed for the bilinear case.

This paper is organized as follows. We begin with a brief review of the well-known case of optimizing a tree

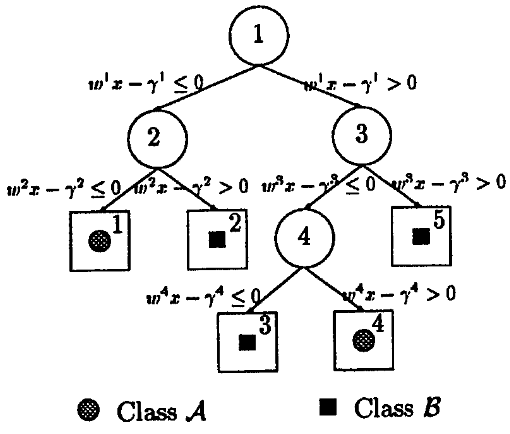


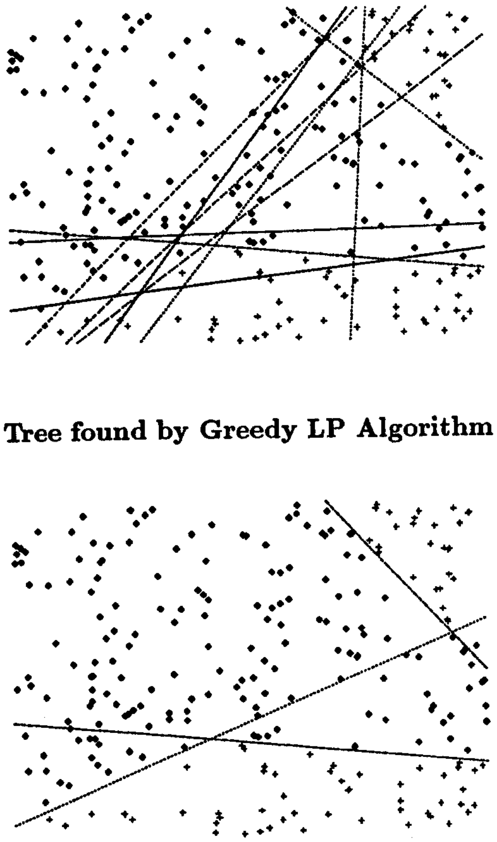
Figure 1: A typical two-class decision tree

consisting of a single decision. The tree is represented as a system of linear inequalities and the system is solved using linear programming. In Section 3 we show how more general decision trees can be expressed as a system of disjunctive linear inequalities and formulated as a multilinear programming problem. Section 4 explains the iterative linear programming algorithm for optimizing the resulting problem. Computational results and conclusions are given in Section 5.

GTO applies to binary trees with a multivariate decision at each node of the following form: If x is a point being classified, then at decision node d, if $xw^d > \gamma^d$ the point follows the right branch, if $xw^d \le \gamma^d$ then the point follows the left branch. The choice of which branch the point follows at equality is arbitrary. This type of decision has been used in greedy algorithms [6, 1]. The univariate decisions found by CART [5] for continuous variables can be considered special cases of this type of decision with only one nonzero component of w. A point is classified by following the path of the point through the tree until it reaches a leaf node. A point is strictly classified by the tree if it reaches a leaf of the correct class and equality does not hold at any decision along the path to the leaf (i.e. $xw^d \neq \gamma^d$ for any decision d in the path). Although GTO is applicable to problems with many classes, for simplicity we limit discussion to the problem of classifying the two sets $\mathcal{A}$ and $\mathcal{B}$. A sample of such a tree is given in Figure 1. Let $\mathcal{A}$ consist of k points contained in R^n and $\mathcal{B}$ consist of m points contained in R^n. Let A_j denote the jth point in $\mathcal{A}$.

2 Optimizing a Single Decision

Many methods exist for minimizing the error of a tree consisting of a single decision node. We briefly review one approach which formulates the problem as a set of linear inequalities and then uses linear programming to minimize the errors in the inequalities [3]. The reader is referred to [3] for full details of the practical and theoretical benefits of this approach.



(a) Tree found by Greedy LP Algorithm

(b) Tree found by GTO

Figure 2: Geometric depiction of decision trees

Let $xw = \gamma$ be the plane formed by the decision. For any point x, if $xw < \gamma$ then the point is classified in class $\mathcal{A}$, and if $xw > \gamma$ then the point is classified in class $\mathcal{B}$. If $xw = \gamma$ the class can be chosen arbitrarily. All the points in $\mathcal{A}$ and $\mathcal{B}$ are strictly classified if there exist w and γ such that

$$\begin{array}{ll} A_j w - \gamma < 0 & j = 1 \ldots m \\ B_i w - \gamma > 0 & i = 1 \ldots k \end{array} \qquad (1)$$

or equivalently

$$\begin{array}{ll} -A_j w + \gamma \geq 1 & j = 1 \ldots m \\ B_i w - \gamma \geq 1 & i = 1 \ldots k \end{array} \qquad (2)$$

Note that Equations (1) and (2) are alternative definitions of linear separability. The choice of the constant 1 is arbitrary. Any positive constant may be used.

If $\mathcal{A}$ and $\mathcal{B}$ are linearly separable then Equation (2) is feasible, and the linear program (LP) (3) will have a zero minimum. The resulting (w, γ) forms a decision that strictly separates $\mathcal{A}$ and $\mathcal{B}$. If Equation (2) is not feasible, then LP (3) minimizes the average misclassification error within each class.

$$\begin{array}{lllll} \min\limits_{w,\gamma} & \frac{1}{m}\sum\limits_{j=1}^{m} y_j + \frac{1}{k}\sum\limits_{i=1}^{k} z_i & & & \\ s.t. & y_j \geq A_j w - \gamma + 1 & y_j \geq 0 & j = 1 \ldots m & \\ & z_i \geq -B_i w + \gamma + 1 & z_i \geq 0 & i = 1 \ldots k & \end{array} \qquad (3)$$

LP (3) has been used recursively in a greedy decision tree algorithm called Multisurface Method-Tree (MSMT) [1]. While it compares favorably with other greedy decision tree algorithms, it also suffers the problem of all greedy approaches. Locally good but globally poor decisions near the root of the tree can result in overly large trees with poor generalization. Figure 2 shows an example of a case where this phenomenon occurs. Figure 2a depicts the 11 planes used by MSMT to completely classify all the points. The decisions chosen near the root of the tree are largely redundant. As a result the decisions near the leaves of the tree are based on an unnecessarily small number of points. MSMT constructed an excessively large tree that does not reflect the underlying structure of the problem. In contrast, GTO was able to completely classify all the points using only three decisions (Figure 2b).

3 Problem Formulation

For general decision trees, the tree can be represented as a set of disjunctive inequalities. A multilinear program is used to minimize the error of the disjunctive linear inequalities. We now consider the problem of optimizing a tree with the structure given in Figure 1, and then briefly consider the problem for more general trees.

Recall that a point is strictly classified by the tree in Figure 1 if the point reaches a leaf of the correct classification and equality does not hold for any of the decisions along the path to the leaf. A point $A_j \in \mathcal{A}$ is strictly classified if it follows the path through the tree to the first or fourth leaf node, i.e. if

$$\begin{array}{c} \left\langle \begin{array}{c} A_j w^1 - \gamma^1 + 1 \leq 0 \\ A_j w^2 - \gamma^2 + 1 \leq 0 \end{array} \right\rangle \\ \textbf{or} \\ \left\langle \begin{array}{c} -A_j w^1 + \gamma^1 + 1 \leq 0 \\ A_j w^3 - \gamma^3 + 1 \leq 0 \\ -A_j w^4 + \gamma^4 + 1 \leq 0 \end{array} \right\rangle \end{array} \qquad (4)$$

or equivalently

$$\begin{array}{c} (A_j w^1 - \gamma^1 + 1)_+ \cdot (A_j w^2 - \gamma^2 + 1)_+ = 0 \\ \textbf{or} \\ (-A_j w^1 + \gamma^1 + 1)_+ \cdot (A_j w^3 - \gamma^3 + 1)_+ \cdot \\ (-A_j w^4 + \gamma^4 + 1)_+ = 0 \end{array} \qquad (5)$$

where $(\zeta)_+ := \max\{\zeta, 0\}$.

Similarly a point $B_i \in \mathcal{B}$ is strictly classified if it follows the path through the tree to the second, third, or fifth leaf node, i.e. if

$$\begin{array}{c} \left\langle \begin{array}{c} B_i w^1 - \gamma^1 + 1 \leq 0 \\ -B_i w^2 + \gamma^2 + 1 \leq 0 \end{array} \right\rangle \\ \textbf{or} \\ \left\langle \begin{array}{c} -B_i w^1 + \gamma^1 + 1 \leq 0 \\ B_i w^3 - \gamma^3 + 1 \leq 0 \\ B_i w^4 - \gamma^4 + 1 \leq 0 \end{array} \right\rangle \\ \textbf{or} \\ \left\langle \begin{array}{c} -B_i w^1 + \gamma^1 + 1 \leq 0 \\ -B_i w^3 + \gamma^3 + 1 \leq 0 \end{array} \right\rangle \end{array} \qquad (6)$$

or equivalently

$$\begin{array}{c} (B_i w^1 - \gamma^1 + 1)_+ \cdot (-B_i w^2 + \gamma^2 + 1)_+ = 0 \\ \textbf{or} \\ (-B_i w^1 + \gamma^1 + 1)_+ \cdot (B_i w^3 - \gamma^3 + 1)_+ \cdot \\ (B_i w^4 - \gamma^4 + 1)_+ = 0 \\ \textbf{or} \\ (-B_i w^1 + \gamma^1 + 1)_+ (-B_i w^3 + \gamma^3 + 1)_+ = 0 \end{array} \qquad (7)$$

A decision tree exists that strictly classifies all the points in sets $\mathcal{A}$ and $\mathcal{B}$ if and only if the following equation has a feasible solution:

$$\begin{array}{l} \sum\limits_{j=1}^{m} (y_{1_j} + y_{2_j}) \cdot (z_{1_j} + y_{3_j} + z_{4_j}) + \\ \sum\limits_{i=1}^{k} (u_{1_i} + v_{2_i}) \cdot (v_{1_i} + u_{3_i} + u_{4_i}) \cdot (v_{1_i} + v_{3_i}) \;=\; 0 \\ \quad where \;\; \begin{array}{ll} y_{d_j} = (A_j w^d - \gamma^d + 1)_+ & j = 1 \ldots m \\ z_{d_j} = (-A_j w^d + \gamma^d + 1)_+ & \\ u_{d_i} = (B_i w^d - \gamma^d + 1)_+ & i = 1 \ldots k \\ v_{d_i} = (-B_i w^d + \gamma^d + 1)_+ & \\ for \; d = 1 \ldots D & \\ \; and \; D \; = \; number \; of \; decisions \; in \; \; tree. & \end{array} \end{array} \qquad (8)$$

Furthermore, (w^d, γ^d), $d = 1 \ldots D$, satisfying (8) form the decisions of a tree that strictly classifies all the points in the sets $\mathcal{A}$ and $\mathcal{B}$.

Equivalently, there exists a decision tree with the given structure that correctly classifies the points in sets $\mathcal{A}$ and $\mathcal{B}$ if and only if the following multilinear program has a zero minimum:

$$\begin{array}{ll}
\min\limits_{w,\gamma,y,z,u,v} & \frac{1}{m}\sum\limits_{j=1}^{k}(y_{1_j}+y_{2_j})\cdot(z_{1_j}+y_{3_j}+z_{4_j})+ \\
& \frac{1}{k}\sum\limits_{i=1}^{m}(u_{1_i}+v_{2_i})\cdot(v_{1_i}+u_{3_i}+u_{4_i})\cdot(v_{1_i}+v_{3_i}) \\
s.t. & y_{d_j}\geq A_jw^d-\gamma^d+1 \quad j=1\dots m \\
& z_{d_j}\geq -A_jw^d+\gamma^d+1 \\
& u_{d_i}\geq B_iw^d-\gamma^d+1 \quad i=1\dots k \\
& v_{d_i}\geq -B_iw^d+\gamma^d+1 \\
& for\ d=1\dots j \\
& y,z,u,v\geq 0
\end{array} \tag{9}$$

The coefficients $\frac{1}{m}$ and $\frac{1}{k}$ were chosen so that (9) is identical to the LP (3) for the single decision case, thus guaranteeing that $w=0$ is never the unique solution for that case [3]. These coefficients also help to make the method more numerically stable for large training set sizes.

This general approach is applicable to any multivariate binary decision tree used to classify two or more sets. There is an error term for each point in the training set. The error for that point is the product of the errors at each of the leaves. The error at each leaf is the sum of the errors in the decisions along the path to that leaf. If a point is correctly classified at one leaf, the error along the path will be zero, and the product of the leaf errors will be zero. Space does not permit discussion of the general formulation in this paper, thus we refer the reader to [2] for more details.

4 Multilinear Programming

The multilinear program (3) and its more general formulation can be optimized using the iterative linear programming Frank-Wolfe type method proposed in [4]. We outline the method here, and refer the reader to [2] for the mathematical properties of the algorithm.

Consider the problem $\min\limits_{x} f(x)$ subject to $x\in\mathcal{X}$ where $f:R^n\rightarrow R$, $\mathcal{X}$ is a polyhedral set in R^n containing the constraint $x\geq 0$, f has continuous first partial derivatives, and f is bounded below. The Frank-Wolfe algorithm for problem is the following:

Algorithm 4.1 (Frank-Wolfe algorithm [7, 4]) *Start with any $x^0\in\mathcal{X}$. Compute x^{i+1} from x^i as follows.*

$$(i) \qquad v^i\in arg\ vertex \min_{x\in\mathcal{X}} \nabla f(x^i)x$$

$$(ii) \qquad Stop\ if\ \ \nabla f(x^i)v^i = \nabla f(x^i)x^i$$

$$(iii) \qquad x^{i+1}=(1-\lambda^i)x^i+\lambda^iv^i\ where$$

$$\lambda^i\in arg\min_{0\leq\lambda\leq 1} f((1-\lambda)x^i+\lambda v^i)$$

In the above algorithm "*arg vertex* min" denotes a vertex solution set of the indicated linear program. The algorithm terminates at some x^j that satisfies the minimum principle necessary optimality condition: $\nabla f(x^j)(x-x^j)\geq 0$, for all $x\in\mathcal{X}$, or each accumulation point $\bar{x}$ of the sequence $\{x^i\}$ satisfies the minimum principle [4].

The gradient calculation for the GTO function is straightforward. For example, when Algorithm 4.1 is applied to Problem (9), the following linear subproblem is solved in step (i) with $(\hat{w},\hat{\gamma},\hat{y},\hat{z},\hat{u},\hat{v})=x^i$:

$$\begin{array}{ll}
\min\limits_{w,\gamma,y,z,u,v} & \frac{1}{m}\sum\limits_{j=1}^{m}(\hat{y}_{1_j}+\hat{y}_{2_j})(z_{1_j}+y_{3_j}+z_{4_j})+ \\
& \frac{1}{m}\sum\limits_{j=1}^{m}(y_{1_j}+y_{2_j})\cdot(\hat{z}_{1_j}+\hat{y}_{3_j}+\hat{z}_{4_j})+ \\
& \frac{1}{k}\sum\limits_{i=1}^{k}(\hat{u}_{1_i}+\hat{v}_{2_i})\cdot(\hat{v}_{1_i}+\hat{u}_{3_i}+\hat{u}_{4_i}) \\
& \qquad\qquad \cdot(v_{1_i}+v_{3_i})+ \\
& \frac{1}{k}\sum\limits_{i=1}^{k}(\hat{u}_{1_i}+\hat{v}_{2_i})\cdot(v_{1_i}+u_{3_i}+u_{4_i})\cdot \\
& \qquad\qquad (\hat{v}_{1_i}+\hat{v}_{3_i})+ \\
& \frac{1}{k}\sum\limits_{i=1}^{k}(u_{1_i}+v_{2_i})\cdot(\hat{v}_{1_i}+\hat{u}_{3_i}+\hat{u}_{4_i})\cdot \\
& \qquad\qquad (\hat{v}_{1_i}+\hat{v}_{3_i}) \\
s.t. & y_{d_j}\geq A_jw^j-\gamma^d+1 \quad For\ d=1,\dots,D \\
& z_{d_j}\geq -A_jw^d+\gamma^d+1 \quad j=1\dots m \\
& u_{d_i}\geq B_iw^d-\gamma^d+1 \quad i=1\dots k \\
& v_{d_i}\geq -B_iw^d+\gamma^d+1 \\
& y,z,u,v\geq 0 \quad fixed\ \hat{y},\hat{z},\hat{u},\hat{v},\geq 0
\end{array}$$

5 Results and Conclusions

GTO was implemented for general decision trees with fixed structure. In order to test the effectiveness of the optimization algorithm, random problems with known solutions were generated. For a given dimension, a tree with 3 to 7 decision nodes was randomly generated to classify points in the unit cube. Points in the unit cube were randomly generated and classified and grouped into a training set (500 to 1000 points) and a testing set (5000 points). MSMT, the greedy algorithm discussed in Section 2, was used to generate a greedy tree that correctly classified the training set. The MSMT tree was then pruned to the known structure (i.e. the number of decision nodes) of the tree. The pruned tree was used as a starting point for GTO. The training and testing set error of the MSMT tree, the pruned tree (denoted MSMT-P), and the GTO tree were measured, as was the training time. This experiment was repeated for trees ranging from 3 to 7 nodes in 2 to 25 dimensions. The results were averaged over 10 trials.

We summarize the test results and refer the reader to [2] for more details. Figure 3 presents the average results for randomly generated trees with three decision nodes. These results are typical of those observed in the other experiments. MSMT achieved 100% correctness on the training set but used an excessive number of decisions. The training and testing set accuracy of the pruned trees dropped considerably. The trees once optimized by GTO were significantly better in terms of testing set accuracy than both unpruned and pruned MSMT trees.

The computational results are promising. The Frank-Wolfe algorithm converges in relatively few iterations to an improved solution. However GTO did not always find the global minimum. We expect the problem to have many local minima since it is NP-complete. We plan to investigate using global optimization techniques to avoid local minima. The overall execution time of GTO tends to grow as the problem size increases. Parallel computation can be used to improve the execution time of the expensive LP subproblems. The LP subproblems (e.g. Problem (9)) have a block-separable structure and can be divided into independent LPs solvable in parallel.

We have introduced a non-greedy approach for optimizing decision trees. The GTO algorithm starts with an existing decision tree, fixes the structure of the tree, formulates the error of the tree, and then optimizes that error. An iterative linear programming algorithm performs well on this NP-complete problem. GTO optimizes all the decisions in the tree, and thus has many potential applications such as: decreasing greediness of constructive algorithms, reoptimizing existing trees when additional data is available, pruning greedy decision trees, and incorporating domain knowledge into the decision tree.

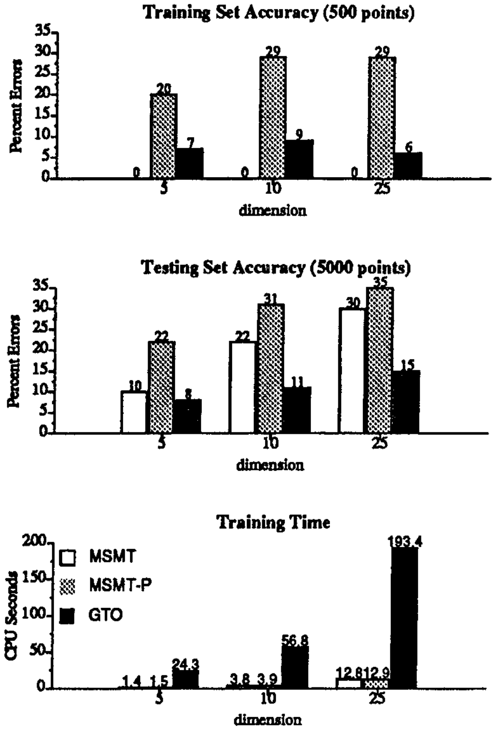


Figure 3: Average results over 10 trials for randomly generated decision trees with 3 decision nodes.

References

[1] K. P. Bennett. Decision tree construction via linear programming. In M. Evans, editor, *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, pages 97–101, Utica, Illinois, 1992.

[2] K. P. Bennett. Optimal decision trees through multilinear programming. R.P.I. Math Report No. 214, Rensselaer Polytechnic Institute, Troy, NY, 1994.

[3] K. P. Bennett and O. L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[4] K. P. Bennett and O. L. Mangasarian. Bilinear separation of two sets in n-space. *Computational Optimization and Applications*, 2:207–227, 1993.

[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees.* Wadsworth International, Californma, 1984.

[6] C. E. Brodley and P. E. Utgoff. Multivariate decision trees. COINS Technical Report 92-83, University of Massachussets, Amherst, Massachusetts, 1992. To appear in *Machine Learning.*

[7] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3:95–110, 1956.

[8] J. H. Friedman. Multivariate adaptive regression splines (with discussion). *Annals of Statistics*, 19:1–141, 1991.

[9] N. Megiddo. On the complexity of polyhedral separability. *Discrete and Computational Geometry*, 3:325–337, 1988.

[10] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1984.

Growing Decision Trees Less Greedily

John F. Elder IV, Rice University

Dept. Computational & Applied Mathematics, Box 1892, Houston Texas 77251 [elder@rice.edu]

Key Words: model selection, regression, decision trees, CART, TX2step, neural networks

Abstract[1]

Most algorithms which induce model structure from sample data proceed, to varying degrees, "greedily". That is, they sequentially add to the current model the candidate component which works best with the existing structure. (Such components include a linear term with stepwise regression, a small polynomial with GMDH-like methods[2], or a threshold split with decision trees.)

This greedy search procedure is relatively fast, but is not optimal, as there can exist models within the "reachable" space which have less complexity and/or greater accuracy on the training data. Indeed, this difference in training performance between optimal and greedy models can be surprisingly large. Still, it is not clear how much greediness hurts in practice, and whether greedy models typically under perform on unseen, but similar data.

Here, we review example effects of greediness in regression to motivate study of the issue with another popular model form: *decision trees*. A new tree algorithm, "Texas Two-Step", is introduced which looks ahead one more generation than standard procedures. In other words, it judges a potential split not by how the resulting child nodes turn out, but by how the grandchildren do. Preliminary results are compared on a recent field application: identifying a bat's species by its chirps.

1. Automated Induction

Inductive algorithms are, at one level, "black boxes" for developing classification, estimation, or control models from sample data. They automatically search a vast space of potential models for the best inputs, structure (terms and interconnections), and parameter values. The models are pieced together in a stepwise manner into a feed-forward network (e.g., tree) of simple nodes. The better methods also *prune* unnecessary terms or nodes from the model, thereby regulating complexity to reduce the chance of *overfit*. Overfit models are over-specialized to the training data and generalize poorly (fail on new data). This is widely held to be the chief danger of using inductive methods.

Complexity is regulated either through

1) *term penalties*, as with model selection criteria such as C_p (Mallows, 1973) and *Minimum Description Length*, MDL (Rissanen, 1978),
2) *roughness penalties* (integrated second derivatives of the estimation surface), or
3) *tests on withheld data* (e.g., *V*-fold cross-validation).

The penalties add to an error measure, and models having the lowest combined score are judged the best candidates for use.

Stepwise regression can be considered a low-level automated induction algorithm. Though the set of possible models (linear combinations of a subset of original candidate inputs) is quite constrained, the procedure does identify which variables to employ and can increase or reduce the size of the set under consideration.

In contrast, *Artificial Neural Networks* (ANNs) are not inductive methods by the definition used here, as their structure is fixed *a priori*.[3] They can more precisely be viewed as a class of nonlinear models whose parameters are typically set through a local gradient search called *back-propagation*.[4] (One suspects that ANNs, which can perform well even when they appear over-parameterized, may avoid overfit partly because of the weakness of this search algorithm! It is possible that improvement of the search procedure without simplification of the model structure may result in better training but worse out-of-sample performance.)[5]

Leading automated induction methods, using "building blocks" consisting of logistic functions, splines, polynomials, planes, non-parametric smoothes of weighted sums, etc. -- are briefly described in (Elder, 1993) along with their chief strengths and weaknesses. Here, we focus on one of the

[1] This work was partially supported by an NSF Research Associateship in Computational Science and Engineering.

[2] Group Method of Data-Handling (Ivakhenko, 1968). See also the book edited by Farlow, 1984.

[3] Removing small terms within ANN nodes does not address over-parameterization, where useless terms can appear significant though their coefficients collectively cancel. (The dangers of collinear variables in regression are analogous.)

[4] This iterative search converges relatively slowly to a local minimum in parameter space, and it has recently been shown (Mulier and Cherkassky, 1993) that the presentation order of the data affects the particular minimum found.

[5] If this danger is real, then the "greedy" nature of the gradient search may have benefits as well.

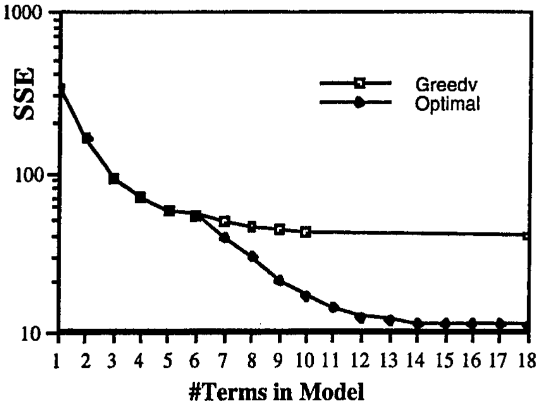


Figure 1: Greedy vs. Optimal Subset Selection

latter: greediness, and look briefly at its effect on regression and decision trees.

2. Subset Selection in Regression

Due to the combinatorial explosion of a trial-and-error search process (the methods are at least polynomial in the inputs and often exponential), a greedy heuristic is often employed: models are constructed in stages, and only the current step is optimized at a given time. *Forward selection* finds the single best term, then adds to it the term which works best with the first, then the one which best assists the pair, and so on. (Note that this is very much more useful than a "first impression" model, which ranks the candidate terms according to their individual performance and employs the top *K*.) *Reverse elimination* begins with a "full" model and sequentially removes the least useful term.

A combined method, *stepwise selection* (e.g., Draper and Smith, 1966) considers removing variables after each new variable is introduced. The standard selection mechanism, checking "F-to-enter" and "F-to-exit" significance values, is a kind of heuristic term penalty method, but not a correct use of F-tests. (The static significance measure is invalid in the dynamic modeling situation and can lead to highly inflated confidences in the resulting parameter values; see, e.g., Miller, 1990).

This greedy growth strategy makes the search feasible and often discovers useful features, but can miss "reachable" structure in the data; that is, within the form of the basis functions employed. For example, given $Y=\{1,1,1,1\}$, $X_1=\{1,1,1,0\}$, $X_2=\{1,1,0,0\}$, $X_3=\{0,0,1,1\}$, a stepwise procedure would first choose x_1 with which to estimate Y, and then seek to add another x. However, an exact model, $Y = x_2 + x_3$, would not include that single best input. Surprisingly, even if there is agreement between the forward and backward procedures on the best model of each size, they can differ by an arbitrarily large amount from some of the best subsets (Berk, 1978).

For example, Desroachers and Mohseni (1984) presented a purportedly optimal algorithm for model selection, and demonstrated it on a problem of estimating rocket engine temperature (from Lloyd and Lipow, 1962), where their small set results agreed with earlier analyses by Draper and Smith (1966). However, the approach turned out to be a version of forward selection. To compare these models with optimal subsets (of the candidate set defined by Desroachers and Mohseni), a new technique for term elimination had to be developed (Elder, 1990). Figure 1 shows the SSE of the greedy and optimal models of each size. The former leveled off at a limit of 40, while the latter were able to reach nearly the minimum error possible for the data (approximated by the *Y* axis base). Clearly, greedy methods can be improved upon significantly, in training, on real applications.

For regression model building, a logical extension of the greedy growth strategy (while stopping short of the hope of "optimal" models) is to add *chunks* of terms at a time, rather than just one. This is the heart the approach taken in GMDH-like techniques, such as ASPN (Algorithm for the Synthesis of Polynomial Networks, Elder, 1985). There, sets of several terms, employing a few independent variables, are considered for inclusion simultaneously, then pared down by reverse elimination. Nodes of such equations are built up until the added complexity cannot be justified, according to a penalty criterion -- either Predicted Squared Error (A. Barron, 1984) or MDL. An ASPN regression network, such as that shown in Figure 2, can have multiple layers of diverse nodes, each with several terms, resulting in a flexible compound function form.

Extensive comparison with more greedy algorithms has yet to be performed, but several researchers have successfully employed such regression networks on applications which had proven very difficult by other methods, including automatic pipe inspection (Mucciardi, 1982), fish stock classification (Prager, 1988), reconfigurable flight control (Elder and Barron, 1988), tactical weapon guidance (Barron and Abbott, 1988), and temperature distribution forecasting (Fulcher and Brown, 1991). Though several areas of possible improvement have been identified (Elder and Brown, 1994), its success suggests that taking complex, rather than

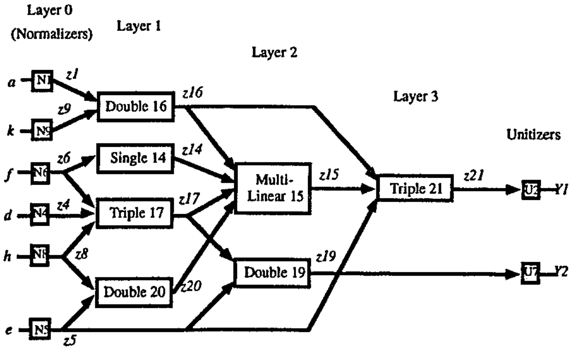


Figure 2: Sample Regression Network

simple, steps might improve other constructive algorithms for induction, such as those used to build decision trees.

3. Constructing Decision Trees

Though there are other and earlier decision tree algorithms (e.g., ID3 and CHAID), CART (Classification and Regression Trees, Breiman, Friedman, Olshen and Stone, 1984) is perhaps the best known and, arguably, most powerful. Some of its nicer features include built-in cross-validation, the ability to handle categorical variables and missing data, and a good presentation of the output. (Versions are also appearing which tie into commercial statistical packages and improve the interface.) Still, the basic classification algorithm is very simple: try to discriminate between classes by recursively bifurcating the data until the resulting groups are as pure as can be sustained. That is, start with all the training data and choose the univariate threshold split (e.g., $x3 < 1.14$) which divides the sample into two maximally pure parts (i.e., minimizes the sample variance of the sum). (Multi-linear splits (e.g., $x1 + 2x2 < 3$) are possible, but do not seem to work well in practice, perhaps because of a poor internal search algorithm.) Then, continue with each of the parts (child nodes) until either no splits are possible, or the leaves (terminal nodes of the tree) are pure (represent only one class) or have some minimum size. Then, CART prunes back (simplifies) the tree, typically using cross-validation, to avoid overfit. This over-training followed by pruning was found by CART's authors to lead to better trees than under the competing method of trying to select the growth stopping point.

For estimation, the leaves are set to the mean or median value of the cases contained, forming a piecewise-constant surface, as shown in Figure 3 for a 4-node tree.

This simple splitting approach is nevertheless powerful, as a sequence of threshold questions quickly conditions an individual case. Each path down the tree can have its own important variables and outliers have no special influence. Also, as with other methods which implicitly select variables, a user can feel free to try more candidates than otherwise, since CART will sift through them unfettered by concerns about multicollinearity, which can hurt regression methods. However, if the candidate variables are jointly useful, relatively independent, and not beset by many outliers, other methods of discrimination can outperform CART.

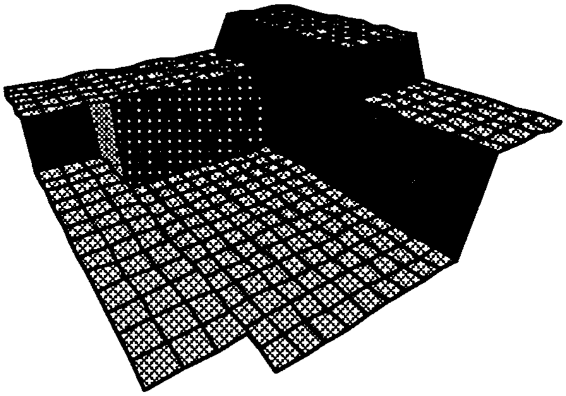


Figure 3: Example Decision Tree Surface

Table 1: Greedy Counter-Example for CART

Y	*a*	*b*	*c*
0	0	0	0
0	1	0	0
1	0	0	1
1	1	0	1
1	0	1	0
1	1	1	0
0	1	1	1
0	1	1	1

Here, we wonder simply if CART's strategy of choosing the greedy split cannot be improved. As a motivating example, consider the XOR-like data of Table 1. CART forms the approximation tree of Figure 4a (using a leaf size limit of ≤ 2 cases). Its greedy search does not find the simpler, exact tree of Figure 4b.

To explore whether an extension of the horizon to two steps ahead would be beneficial, a decision tree algorithm called "Texas Two-Step" was written.

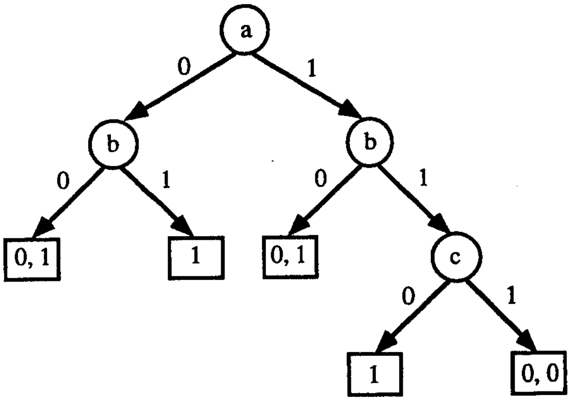


Figure 4a: Inaccurate Greedy Decision Tree

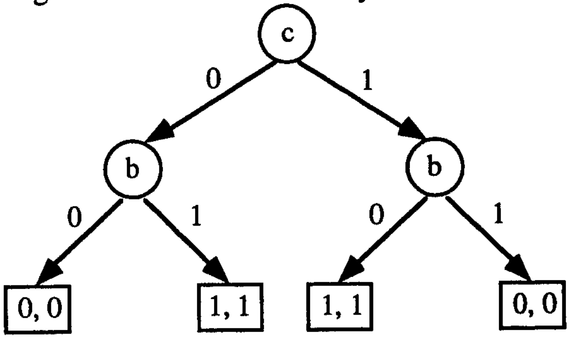


Figure 4b: Correct Decision Tree

4 Texas Two-Step (TX2step)

The algorithm TX2step is a slimmed-down version of CART for classification which is not able to handle missing data, perform internal cross-validation, set misclassification costs, or adjust priors, and so on. Yet it can look two steps ahead to choose the current split, and thereby finds the tree of Figure 4b given the data of Table 1. TX2step has one other new feature:: given more than one split which results in the same score, it uses the split with the *largest relative gap* between border training cases. That is, the tie-breaker to choose the dimension *d* of the split depends on

$$\text{gap}[d] = 0.5\,\frac{\min \text{Right}\,Xd - \max \text{Left}\,Xd}{\max Xd - \min Xd}$$

The algorithm can optionally be greedy as well; in that mode, and ignoring gaps, it was validated on several test problems to reproduce the same tree as CART without cross-validation. Therefore, to focus solely on the greediness issue, TX2step-1 (with gap measurement) was actually run in place of CART on the example application shown next. Training was performed until all nodes were pure, but those leaves with a majority class having <3 cases were pruned back (i.e., re-absorbed into their parent node).

5 Example: Identifying Bat Species

Researchers from the University of Illinois, Urbana/Champaign[6] have measured bat echolocation calls and extracted time-frequency features from the signals, toward developing an automated classifying system to track species of bats -- especially those considered endangered. After visualization of projections of the data by the author, and analysis of correlations, multicollinearity, redundancy, and outliers (for suggested techniques see e.g., Elder, 1993), some variables were eliminated and other new ones tried at UIUC, resulting in a database of 93 cases, each with 15 candidate input features, representing 5 different species (classes) of bats.[7,8] One of the better projections of the data is shown in Figure 5, where the classes are noted by different symbols. Note that the groups do tend to cluster but that a fair amount of overlap is evident in this (and all low-d) views.

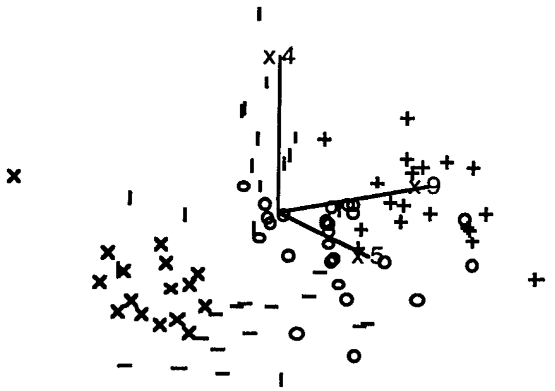


Figure 5: Example Projection of Bat Classes

Trained on all the data, the 1-step tree, shown in Figure 6, had 5 splits (17 prior to pruning) and made 13 training errors. (In the trees, "Yes" answers travel to the left child; "No" to the right.) The 2-step tree of Figure 7 started out simpler, with 14 splits, but pruned less, ending with 10 splits and only 5 training errors. The best root node split happened to be greedy but several other splits were not. For example, the data in the right child node of the root, shown in Figures 8 and 9, are those 58 of 93 cases where *x5* > 101.5. The greedy tree was drawn to split first on *x20* <3.59, then on *x4* < 44.5, and it missed 6 cases on that branch. The 2-step tree instead first chose *x11* < 0.39 -- a seemingly worse split, but when followed by *x4* < 43.5 on one branch, one which allowed it to correctly classify 4 more cases. (The difficulties the split caused its sibling branch were cleared up by subsequent splits.) The 2-step cuts were often more appealing visually; that is, they

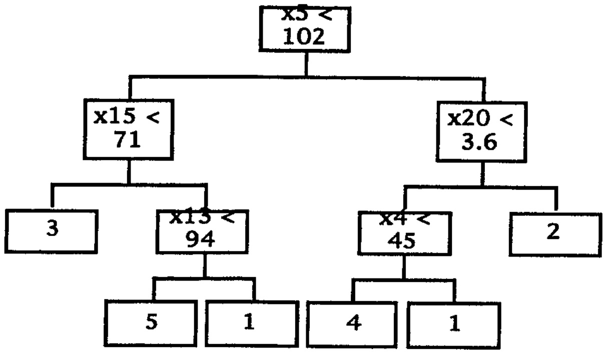


Figure 6: CART (1-step) Tree (using all data)

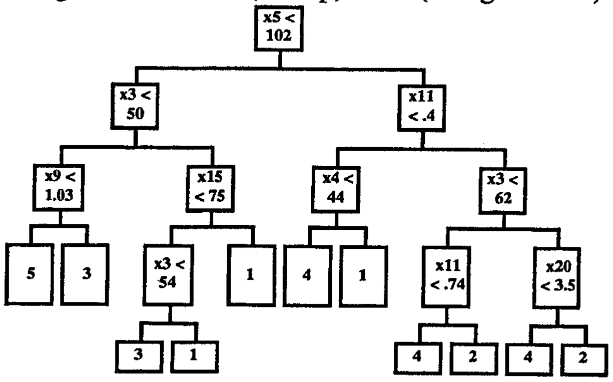


Figure 7: TX2step Tree (using all data)

[6] Biologists Ken White, Curtis Condon, and Al Feng, and Electrical Engineers Oliver Kaefer and Doug Jones.

[7] A single bat from a sixth "Long-Eared" species contributed 5 signals originally, but was removed since it could be easily distinguished by its low-frequency signals and since having only one representative did not allow proper evaluation testing.

[8] It takes less than a second on a SPARC-2 to run the 1-step algorithm on this problem, but about 75 seconds for 2 steps.

accorded more with what an analyst would do when viewing two dimensions of data simultaneously, rather than one.

As expected, the less greedy algorithm performed better on training data. The best test, of course, involves new data. Since there were not many cases, a cross-validation evaluation was performed, where all 3-8 signals for each bat, in turn, were held out of training and independently run down the tree for testing (18 runs for each method). Tables 2-4 show the resulting *confusion matrices* for CART, TX2step, and a neural network (courtesy of Oliver Kaefer and Doug Jones of UIUC) trained on the variables selected by the two tree methods. Correct classifications are along the diagonal and the hit percentage is shown in the corner.

CART gets 43 of 93 signals correct (46%), TX2step 54 (58%), and the ANN performs best with 64 (69%). The difference in accuracy for the tree methods appears more critical when using a *voting scheme*, where several different signals from a single bat are classified and the majority class is assigned. Then, CART misses 11 of the 18 bats but TX2step only 6. (The voting ANN misses just 4.)

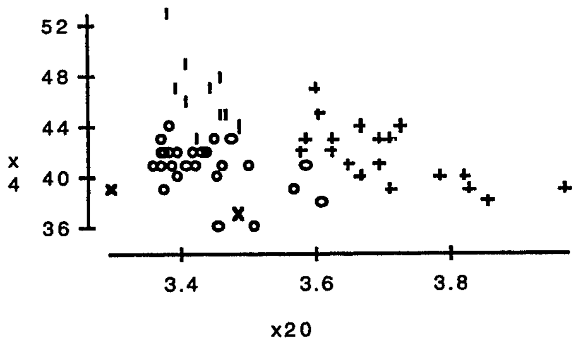


Figure 8: CART View at Right Node

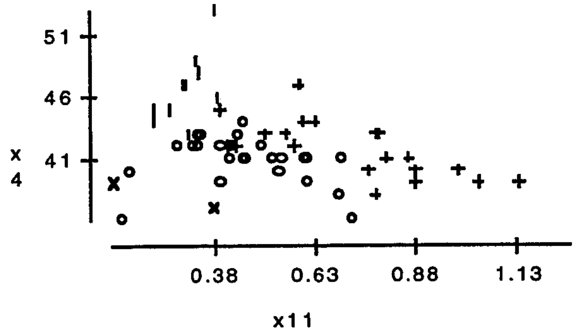


Figure 9: TX2step View at Right Node

In this experiment (counter to our usual experience), the tree methods were outperformed by an ANN. However, the variable selection performed by CART and TX2step proved helpful to the ANN; one trained on all 35 original data features got only 52% correct in bat-wise cross-validation, and one trained on 17 variables (those given as candidates to the tree methods) was 63% correct. Here, simpler ANNs performed better on new data. Clearly, an inductive ANN algorithm, which adapts the network structure to the data, would be a useful tool. The data characteristics -- filtered features, lack of outliers, clustered classes -- which helped the neural network perform well, should also be agreeable to exemplar-based statistical techniques, such as *kernels* and *nearest neighbors*. (We hope to soon try them, as well as regression networks and other inductive methods.)

Confusion Matrices

Table 2: CART

		True Class 1	2	3	4	5	Tot
P C	1	7	2	2	4	3	18
R L	2		16	4	2		22
E A	3	2		2		6	10
D S	4	5	1		13	2	21
. S	5	4		6	7	5	22
	Tot	18	19	14	26	16	46%

Table 3: TX2step

		True Class 1	2	3	4	5	Tot
P C	1	7		1	7	2	17
R L	2		15		3		18
E A	3	3		9		5	17
D S	4	6	4		16	2	28
. S	5	2		4		7	13
	Tot	18	19	14	26	16	58%

Table 4: 8-Input Neural Network

		True Class 1	2	3	4	5	Tot
P C	1	6	1	1	5	1	14
R L	2	2	16	1	3		21
E A	3	2		11		2	19
D S	4	5	2		18		25
. S	5	3		1		13	14
	Tot	18	19	14	26	16	69%

6 Performance on New Data: Remarks

We have seen that regression subsets and decision trees can be sub-optimal if the single best step is always taken. This is true in other venues as well. Cover (1974) showed an investigation in which greed hurts, where: If only one experiment is allowed, E_1 provides the most information, but if two are possible, then independent versions of the "worse" experiment E_2 are better.

But the degree to which greediness generally hurts performance in practice, on new data, is an open question. Berk (1978) sounded a slightly cautionary note in the case of regression subset selection. Using nine well-studied data sets (having from 4 to 15 predictors, 13 to 541 cases, and often more analysts!), he noted the *maximum* training error difference between all-subsets (optimal) models and both 1) forward selection and 2) reverse elimination models. An improvement of up to 29% in SSE was observed. Then, the sample distributions of each data set were

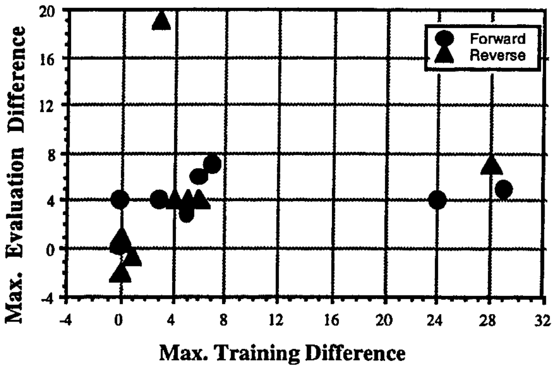


Figure 10: Ex. Training vs. Evaluation Improvement of Optimal over both Forward and Reverse Greedy Methods

employed to generate synthetic data with known population characteristics, and the study again performed for this new evaluation data. Figure 10 plots the training vs. evaluation data differences for the forward and reverse models from the (Berk, 1978) study. Most evaluation differences were smaller and in a tighter range (-2 to 7%, with one exception). In two cases, a greedy method won on the evaluation data by a slight margin.

Note that the differences are somewhat exaggerated, as the maximum disagreement between methods is shown, not that at some automated stopping point. For instance, the two worst reverse values (one training, one evaluation), are for models of size 1 and 2 -- where the forward method would clearly be preferable. Still, the greedy training and evaluation under-performances are correlated, and it can tentatively be concluded that regression differences on new data, while usually less dramatic than on training data, are still likely to be significant.

This was also shown to be the case for decision trees, where a version of CART was out-performed on an example problem by TX2step, which looks ahead an additional step when selecting a threshold for the current node. Further research is planned to examine the effects of greedy model construction strategies in these and other inductive methods, with the hope of understanding better the trade-offs between complexity (in the algorithm as well as model) and accuracy (training and evaluation).

References

Barron, A.R. (1984). Predicted Squared Error: A Criterion for Automatic Model Selection. Ch. 4 (Farlow, 1984)

Barron, R.L. and D. Abbott (1988). User of Polynomial Networks in Optimum, Real-time, Two-Point Boundary Value Guidance of Tactical Weapons, *Proc. Military Comp. Conf.*, Anaheim, CA, May 3-5.

Berk, K.N. (1978). Comparing Subset Regression Procedures, *Technometrics* **20**, no. 1: 1-6.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone (1984). *Classification and Regression Trees.* Wadsworth & Brooks, Pacific Grove, CA.

Cover, T.M. (1974). The Best Two Independent Measurements Are Not the Two Best. *IEEE Trans. Systems, Man & Cybernetics* **4**.

Desroachers, A. and S. Mohseni (1984). On Determining the Structure of a Non-Linear System, *International Journal of Control* **40**: 923-938.

Draper, N.R. and H. Smith (1966). *Applied Regression Analysis.* Wiley, New York.

Elder, J.F. IV (1985). *User's Manual: ASPN: Algorithm for Synthesis of Polynomial Networks* (4th Ed., 1988). Barron Assoc. Inc., Stanardsville, VA.

Elder, J.F. IV (1990). Feature Elimination Using High-Order Correlation, *Proc. Aerospace Applications of Artificial Intelligence,* Dayton, OH, Oct 29-31:65-72.

Elder, J.F. IV (1993). Assisting Inductive Modeling through Visualization, *Proc. Joint Statistical Mtg.*, San Francisco, CA, Aug. 7-11.

Elder, J.F. IV and R.L. Barron (1988). Automated Design of Continuously-Adaptive Control: The "Super-Controller" Strategy for Reconfigurable Systems, *Proc. American Control Conf.*, Atlanta, GA, June 15-17.

Elder, J.F. IV, D.E. Brown (1994, to appear). Induction and Polynomial Networks, in *Advances in Control Networks and Large Scale Parallel Distributed Processing Models* Vol. 2. Ablex, Norwood, NJ (avail. from Univ. VA, Charlottesville, as IPC-TR-92-9).

Farlow, S.J. (1984), Ed. *Self-Organizing Methods in Modeling: GMDH Type Algorithms.* Marcel Dekker.

Fulcher, G.E. and D.E. Brown (1991). A Polynomial Network for Predicting Temperature Distributions, Institute for Parallel Computation Tech. Rpt. 91-008, Univ. VA, June.

Ivakhnenko, A.G. (1968). The Group Method of Data Handling -- A Rival of the Method of Stochastic Approximation, *Soviet Automatic Control* **3**.

Lloyd, D.K., and M. Lipow (1962). *Reliability: Manangement, Methods, and Mathematics.* Prentice Hall, Englewood Cliffs: 360.

Mallows, C.L. (1973). Some Comments on C_p, *Technometrics* **15**: 661-675.

Miller, A.J. (1990). *Subset Selection in Regression.* Chapman and Hall, NY.

Mucciardi, A.N. (1982). ALN 4000 Ultrasonic Pipe Inspection System. *Nondestructive Evaluation Program: Progress in 1981*, EPRI Rpt. NP-2088-SR, Jan.

Mulier, F., V. Cherkassky (1993). *Statistical Analysis of Self-Organization*, Dept. EE, Univ. Minnesota, Minneapolis, MN 55455.

Prager, M.H. (1988). Group Method of Data Handling: A New Method for Stock Identification. *Trans. American Fisheries Society* **117**: 290-296.

Rissanen, J. (1978). Modeling by Shortest Data Description, *Automatica* **14**: 465-471.

Tree Structured Density Estimation

Clifton D. Sutton
George Mason University

Abstract

Tree structured density estimates are produced via a technique which is similar to CART's tree growing algorithm. Various splitting rules are investigated and both the univariate case and the multivariate case are considered. For high-dimensional densities, determining the prominent features of the density through an examination of a binary tree structured estimate is an alternative to attempts at direct visualization of estimates constructed using kernals and other methods.

1 Introduction

Suppose that it is desired to estimate the pdf $f(x_1, \ldots, x_d)$ of the d-dimensional random variable $\vec{X} = (X_1, \ldots, X_d)$, where $d \geq 1$. One could begin by letting $A_1, \ldots, A_m$ be a partition of the sample space and estimating the average density for each set in the partition. Letting v_i be the content (or volume) of A_i, we have

$$v_i = \int \cdots \int_{A_i} dx_1 \ldots dx_d.$$

The average density for A_i is defined by

$$f_i = v_i^{-1} \int \cdots \int_{A_i} f(x_1, \ldots, x_n)\, dx_1 \ldots dx_d$$

and so

$$f_i = P(\vec{X} \in A_i)/v_i.$$

Letting N_i $(i = 1, \ldots, m)$ be the number of observations in a random sample of size n which belong to A_i, we have that

$$\hat{f}_i = \left(\frac{N_i}{n}\right)/v_i$$

is an unbiased estimator for f_i which converges to f_i with probability 1.

Now consider the problem of estimating $f(\vec{x}^*)$, for some $\vec{x}^*$ belonging to the sample space. If f is everywhere continuous in a neighborhood containing $\vec{x}^*$, then by considering a suitably fine partition, f^* can be made to be arbitrarily close to $f(\vec{x}^*)$, where $f^* = f_i$ iff $\vec{x}^* \in A_i$ (i.e., f^* is the average density for the set in the partition to which $\vec{x}^*$ belongs). So if the partition is sufficiently fine and the sample size is sufficiently large, then the density estimator

$$\hat{f}(\vec{x}) = \sum_{i=1}^{m} \hat{f}_i I_{A_i}(\vec{x}),$$

which assigns a constant value to all points belonging to a given set in the partition, should perform well.

If we have a finite sample size, the problem of selecting a good partition to use for the density estimator $\hat{f}$ given above is an interesting one. Even for the simplest case of $d = 1$, the problem of determining the best partition on which to construct a histogram estimator has received considerable attention. Scott [7] reviews various rules which have been suggested for choosing the bin width for fixed bin width histogram estimators, including those of Sturges [8], Scott[6], Freedman and Diaconis [4], and others. Histograms constructed using adaptive meshes (i.e., the selection of a constant bin width is eschewed in favor of a data-based method of creating bins of unequal width) have been considered by Wegman [10,11], Van Ryzin [9], and others, but Scott [7] warns that in practice caution should be taken in using adaptive methods. Compared to the $d = 1$ case, much less is known about histogram estimators for $d \geq 2$.

Besides histograms, other methods have been suggested for density estimation, among them frequency polygons, average shifted histograms, and kernal estimators. Assuming that the chief purposes of constructing a density estimate are to determine key features of the density, discern the general nature of the relationships among the variables, and develop some idea about what the density "looks like", as opposed to merely wanting to know the value of $f(\vec{x})$ at one or more particular points in the sample space, a drawback associated with all of these methods is the difficulty in visualizing or effectively summarizing the resulting estimate if d is greater than 2, or perhaps 3. Computer graphics methods incorporating features such as slicing, contour shells, rotation, color, and stereo effect can certainly help, but for high-dimensional cases it can still be very difficult to even get a rough idea about the overall structure of the estimate.

The suggestion put forth here is to construct a d-dimensional histogram estimate having the form given by

$\hat{f}(\vec{x})$ above, using an irregular adaptive partition consisting of d-dimensional hyper-rectangles created by a recursive partitioning scheme which is similar to the method used by CART (see Breiman et al. [1]) for growing classification and regression trees. At each stage in the creation of the partition, an attempt will be made to best divide a hyper-rectangle into two hyper-rectangles for which the average densities differ and for which the division leads to an improved density estimate. By examing the splits leading to the final estimate, noting on which variables and which values the splits are made and how the average density estimates differ in the hyper-rectangles created, it might be possible to determine relationships among the variables which may be otherwise difficult to detect. By inspecting the density estimates for the terminal nodes, it will be easy to locate the regions of high density in the sample space. So the goal will be to create a useful density estimate, having the simple form of a binary tree, which will allow one to detect the salient features of the density without attempting to directly visualize the estimate.

2 Tree Growing Methodology

For simplicity, it will be assumed that the density to be estimated has a compact support which is contained within a d-dimensional hyper-rectangle that will be taken to be the initial partition used in the tree growing process. (If the density does not have compact support, then one can choose an initial hyper-rectangle that contains the convex hull of the data and use the procedures described below to produce an estimate of the conditional density of $\vec{X}$, given that $\vec{X}$ belongs to the hyper-rectangle.) The tree structured density estimate will be produced by recursively partitioning the initial hyper-rectangle, dividing each new hyper-rectangle which is created until there is insufficient evidence to warrent further divisions.

At each step in the tree growing procedure, a hyper-rectangle in the existing partition of the sample space is split into two hyper-rectangles. The quality of the density estimate produced will depend heavily on the method used to determine the variable on which the split should be made and the exact location of the split (the value of the selected variable that corresponds to the division into two hyper-rectangles). The various rules considered below are all based on the same general principle: to select the split from among all candidates under consideration which provides the strongest evidence that the density is nonconstant over the set in the partition to be split.

With all of the rules, the location of the splits will be based on the empirical marginal distributions formed from the sets of observations belonging to each set of the existing partition. At each step in the tree growing process, there will be d conditional marginal distributions associated with a particular partition set to consider and the split ultimately selected will be the "strongest" of all of the splits which can be made based on the d empirical distributions, provided that this best split is strong enough. Therefore, it will suffice to develop splitting rules for univariate random samples, have associated ways of comparing the strengths of splits made on different samples, and determine criteria with which to assess the strength of the strongest split. Below, I will describe several methods of choosing a split point based on a set of values $x_1, \ldots, x_n$ belonging to the interval $(a, b]$ and associated ways to characterize the strength of the split point candidates. For $d \geq 2$, the procedure for selecting and assessing the overall best split is given as well. Note that although conditional distributions are used to determine the splits, the final density estimate needs to be based on the original full sample, using $\hat{f}_i = n_i/(nv_i)$.

The *sample median method* prescribes that the interval $(a, b]$ be split if the location of the sample median is inconsistent with the hypothesis that the conditional density is constant on $(a, b]$. That is, if the location of the sample median differs significantly from $(a+b)/2$, it will be concluded that the conditional density is not uniform on $(a, b]$ and the interval will be split at the sample median into two intervals, one having an estimated density higher than the other one. For the case of n being odd, an assessment of whether or not the location of the sample median provides strong evidence against a uniform conditional density on $(a, b]$ can be based on the value of

$$P(|M - (a+b)/2| \geq c),$$

where M is the $((n+1)/2)$th order statistic from a uniform $(a, b]$ distribution and c is the value of the observed difference $|x_{((n+1)/2)} - (a+b)/2|$. Using a normal approximation, the above probability is about

$$2\Phi\left(-\sqrt{\frac{nc}{b-a-c}}\right).$$

Thus, as a measure of the strengths of the various candidates for the splits, we can use

$$z = \sqrt{\frac{nc}{b-a-c}}$$

and select the split which maximizes this value, where we consider all d variables, for each case letting n be the number of observations in the partition set and letting a and b be the endpoints of the hyper-rectangle corresponding to variable under consideration. Although a modification should be made if n is even, the adjustment will be slight unless n is rather small and so in pratice the z-score above is used for all split candidates.

If the maximum value of z is greater than some critical value $z_{\alpha'/2}$, then the split is made and the search for the next split begins. If α' is taken to be

$$1-(1-\alpha)^{1/d},$$

then a decision to split the hyper-rectangle based on the maximum value of z corresponds to a decision to reject with a size α test of the null hypothesis that the conditional density of $\vec{X}$ is the joint density of d independent uniform random variables against the general alternative.

Letting

$$g_i = \frac{x_{(i)} + x_{(i+1)}}{2} \quad (i = 1, \ldots, n-1),$$

g_i will be called the ith gap point. If the location of the sample median in $(a, b]$ results in the decision to create a split, the split will be made at g_j, where $j = n/2$ if n is even and j is either $(n-1)/2$ or $(n+1)/2$ if n is odd.

If the partition set under consideration contains a mode, then it may be that none of the d sample medians will be very far from the center of the hyper-rectangle even though the joint density is not constant. In order to prevent the recursive partitioning from terminating prematurely with such a partition set, a trial split can be made. If either of the two hyper-rectangles which result from the trial split produce a sufficiently strong split, then the trial split is accepted and the search for further splits continues. Otherwise, the trial split is not retained and the tree growth terminates in that region of the sample space.

The *all possible split points method* considers many possible split points for each interval. The set of $n-1$ gap points will be taken to be the split point candidates. The strength of the split for the candidate $s \in (a, b]$ is based on the proportion of observations which lie in $(a, s]$. If the proportion differs significantly from $\frac{s-a}{b-a}$, which is the expected value of the proportion if the conditional density is uniform over $(a, b]$, then it is concluded that the average density for $(a, s]$ is different from the average density for $(s, b]$. The strength of the evidence in support of the split is measured by

$$z = \frac{|t_s - np_s| - \frac{1}{2}}{\sqrt{np_s(1-p_s)}},$$

where t_s is the number of observations in $(a, s]$ and $p_s = (s-a)/(b-a)$.

When considering the strength of the strongest split overall using the z-score given above, in addition to the simultaneous inference phenomenon due to the fact that more than one marginal distribution is being examined, it is now the case that many possible split points are being considered for each marginal distribution. One might think that this additional source of multiple comparisons can be accounted for by determining if a one-sample Kolmogorov-Smirnov goodness-of-fit test for the uniform distribution produces a significant result, but in fact there is a discrepancy since the Kolmogorov-Smirnov test depends on the maximum value, for all $s \in (a, b]$, of $|t_s - np_s|$, which is not equivalent to assessing the hypothesis of a constant density using the maximum value of the z-score above. This suggests yet another tree growing procedure, called the *Kolmogorov-Smirnov method*, for which the one-sample Kolmogorov-Smirnov statistic is computed based on each of the d empirical marginal distributions and if the largest of these values is sufficiently large (say, corresponding to a rejection of the null hypothesis of a uniform distribution with a size α' test), then the hyper-rectangle is split.

The split will be made on the variable which produces the largest value of the K-S test statistic. The split will be made at the gap point g_j which maximizes

$$|\hat{F}_X(g_j) - F_{unif}(g_j)|,$$

where $\hat{F}_X$ is the empirical cdf and

$$F_{unif}(g_j) = (g_j - a)/(b-a).$$

It is interesting that maximizing $|\hat{F}(g_j) - F_{unif}(g_j)|$ is equivalent to maximizing

$$\int_a^b |\hat{f}_{g_j}(x) - (b-a)^{-1}|dx,$$

where $\hat{f}_{g_j}(x)$ is the piecewise constant density estimate

$$\frac{t_{g_j}}{n(g_j - a)} I_{(a, g_j]}(x) + \frac{n - t_{g_j}}{n(b - g_j)} I_{(g_j, b]}(x)$$

and t_{g_j} is the number of observations in $(a, g_j]$. Thus splitting at the value for which the empirical cdf differs the most from the cdf of a uniform $(a, b]$ random variable corresponds to selecting the two-bin histogram density estimate which differs the most, in an L_1 sense, from the density of a uniform $(a, b]$ random variable.

The *squared difference method* is similar to the K-S method in that the decision of whether or not a split should be made is based on the the value of a test statistic for a goodness-of-fit test. The null hypothesis that the density is constant over $(a, b]$ is tested against the general alternative using the Cramér-von Mises type of statistic given by

$$Q = \int_a^b \left[\hat{F}_X(x) - \frac{x-a}{b-a}\right]^2 \frac{dx}{b-a}.$$

A split is made on the variable which produces the largest of the observed values of Q, provided that the value exceeds the upper level α' critical value of the distribution free statistic. The split will be made at the gap point g_j which is associated with the two-bin histogram density estimate which differs the most, in an L_2 sense, from the density of a uniform $(a, b]$ random variable. That is, g_j is selected to maximize

$$\int_a^b \left(\hat{f}_{g_j}(x) - (b-a)^{-1}\right)^2 dx$$

$$= \frac{\left[t_{g_j}(b-a) - n(g_j-a)\right]^2}{n^2(b-a)(b-g_j)(g_j-a)}.$$

The *likelihood ratio method* is based on a generalized likelihood ratio test of the null hypothesis that the conditional density is constant on $(a, b]$ against the alternative that the density has the form

$$hI_{(a,s]}(x) + h'I_{(s,b]}(x),$$

where

$$h' = \frac{1-(s-a)h}{b-s},$$

$s \in (a, b]$, and $h \neq (b-a)^{-1}$. A rejection of the null hypothesis supports the conclusion that the density over $(a, b]$ is better estimated by splitting the interval into two pieces and estimating the density for each piece seperately with a constant than it is by not splitting the interval and using only one value for the density over $(a, b]$.

For a given value of s, the likelihood function for the sample $x_1, \ldots, x_n$ is maximized by letting

$$h = \frac{t_s}{n(s-a)},$$

where t_s is the number of observations belonging to $(a, s]$. It follows that to maximize the likelihood over both parameters, it is necessary to find the value of s that maximizes

$$\left[\frac{t_s/n}{s-a}\right]^{t_s} \left[\frac{1-(t_s/n)}{b-s}\right]^{n-t_s}.$$

The likelihood ratio is

$$\lambda = \left[\frac{n(b-\hat{s})}{(n-t_{\hat{s}})(b-a)}\right]^n \left[\frac{(n-t_{\hat{s}})(\hat{s}-a)}{t_{\hat{s}}(b-\hat{s})}\right]^{t_s}$$

and the null hypothesis is rejected whenever λ is sufficiently small. The regularity conditions required to insure that the null distribution of $-2\log\lambda$ is asymptotically χ^2_2 are not satisfied for the testing situation under consideration. Nevertheless, I found that a splitting criterion based on the χ^2_2 distribution works satisfactorily. For each variable, the function given for λ above was maximized over all choices of $\hat{s} \in \{g_1, \ldots, g_{n-1}\}$. Letting λ' be the largest such value obtained with all of the variables, a split is made at the maximizing gap point if $-2\log\lambda' \geq \chi^2_{2,\alpha'} = -2\log\alpha'$.

For all of the methods described above, gap points were removed from consideration as split points if a split at the gap point would result in a hyper-rectangle being created which did not contain at least a minimum number of observations. Values considered for this minimum ranged from 3 to 50, but perhaps using a value less than 3 will improve the accuracy of the estimate for the tails of the distribution.

The *regression tree method* of creating a tree structured density estimate makes direct use of CART's procedure for constructing a regression tree and is rather different from the methods previously discussed. First, a regular partition is created by dividing the initial hyper-rectangle into a large number of identically shaped small hyper-rectangles. Next, the number of observations in each small hyper-rectangle is determined and the corresponding density estimate for the partition set is assigned as the y value corresponding to the $\vec{x}$ located at the center of the hyper-rectangle. CART's regression procedure is then used to construct a regression tree based on these $(y, \vec{x})$ values. The partition for this regression tree serves as the partition for the density estimate. Thus, the density estimate based on the initial fine partition is smoothed by CART's regression algorithm to produce an estimate based on a coarser (and most likely irregular) partition.

A nice feature of the regression tree method is that CART's cross-validation procedure can be easily invoked to select the right sized tree. In general, cross-validation could be used in conjunction with the other methods as well. The criterion parameter α which partially governs tree size can be chosen to be large and the minimum number of observations allowed in a partition set can be made small, so that too complex of a tree is first constructed. Then a cross-validation based pruning procedure can be performed to select the tree which corresponds to the most honest estimate, using the estimated likelihood function

$$\prod \hat{f}_{\vec{X}}(\vec{x})$$

to judge accuracy (however, this may not be satisfactory unless it were the case that the estimated likelihood function should be nonzero over the convex hull of the data).

Using CART's regression algorithm also allows for an easy implementation of linear combination splits, although this may make it more difficult to identify the key features of the density with a quick inspection of the tree. Some of the other methods described above are not easily modified to handle linear combination splits, but modifications of the Kolmogorov-Smirnov method and the squared difference method can be considered. Splits can be made which produce the greatest overall difference between the single constant estimate based on the partition set under consideration and the two constant estimate which would result from a split, where the integrated absolute difference or the integrated squared difference can be used as a measure of overall difference.

3 Results

C programs were written to compute density estimates based on the methods described above. A performance study was done using samples of non-uniform pseudo random variates, created using standard techniques (such as those described in Dagpunar [2], Devroye [3], and Knuth [5]). Numerous samples were used, based on combinations of normal, beta, and gamma distributions and having $1 \leq d \leq 3$ and $1000 \leq n \leq 4000$. The well-known Buffalo snowfall data set (having $n = 63$) was also considered. Values used for α ranged from 0.005 to 0.1. Usually, the choice of α had little effect on the density and letting α equal 0.05 seems to be a reasonable choice (but additional study is warrented here). Overall, it appears that the best trees are created when the minimum number of observations allowed in a partition set is more responsible for the size of the tree than is the value of α.

In general, most of the methods tended to produce very similar results in a lot of the cases considered. But not all of the methods have been throughly investigated and so any conclusions are tentative at this time. For the univariate cases, the tree structured estimates were typically quite a bit coarser then histograms constructed from the same samples using some of the common fixed bin width rules. While they might have an overall lack of accuracy based on a criterion such as the MISE (see Scott [7], p. 38), the tree structured estimates very rarely indicated more modes than what was proper, provided that partition sets containing only a small number of observations were disallowed. Furthermore, for all values of d, if partition sets containing only a small number of observations are disallowed, then the average densities for the partition sets (the f_i) were often very closely estimated by the $\hat{f}_i$. In general, the $\hat{f}_i$ were highly correlated with the f_i and in almost every case considered the partition set having the largest estimated average density was the partition set having the largest average density. So, all in all, the tree structured estimators seem to be very good with regard to finding modes. Also, although the coarseness contributes to an overall lack of accuracy, for large d, a finer partition may correspond to a tree structured estimate from which it would be more difficult to identify the main features of the density.

References

[1] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *CART: Classification and Regression Trees.* Belmont, CA: Wadsworth.

[2] Dagpunar, J. (1988). *Principles of Random Variate Generation.* Oxford: Oxford University Press.

[3] Devroye, L. (1986). *Non-Uniform Random Variate Generation.* New York, NY: Springer-Verlag.

[4] Freedman, D. and Diaconis, P. (1981). On the histogram as a density estimator: L_2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete* 57, 453–476.

[5] Knuth, D. E. (1981). *The Art of Computer Programming* (Vol. 2 / Seminumerical Algorithms, 2nd ed.). Reading, MA: Addison-Wesley.

[6] Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika* 66, 605–610.

[7] Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization.* New York, NY: Wiley.

[8] Sturges, H. A. (1926). The choice of a class interval. *Journal of the American Statistical Association* 21, 65–66.

[9] Van Ryzin, J. (1973). A histogram method of density estimation. *Communications in Statistics* 2, 493–506.

[10] Wegman, E. J. (1970). Maximum likelihood estimation of a unimodal density function. *Annals of Statistics* 41, 457–471.

[11] Wegman, E. J. (1975). Maximum likelihood estimation of a probability density function. *Sankhyā* 37, Series A, Pt. 2, 211–224.

Tree-Structured Density Estimation and Dimensionality Reduction

Nong Shang
School of Public Health
University of California, Berkeley
Berkeley, CA, 94720

Abstract

Due to the "curse of dimensionality" and the expression difficulties, high-dimension density estimation is usually an ill-defined problem. However, many practical problems involve in issues of estimating high-dimension densities. Certain kind of dimensionality reduction is then necessary. A computation extensive density estimation method is developed based on the tree-structured methodology. The method has the ability to identify noises in the density structure, therefore greatly reduces the dimensionality. It also provides a simple way to present a high-dimensional density, thus helps us to explore the data structure. Simulation studies show that the method is often more accurate than other regular methods when the density structure is complex.

1 Introduction

The problem of density estimation is to construct a function from a random sample to approximate the real density. The purposes of density estimation is to present and explore the density structure as well as to obtain accurate estimation for other applications. A survey of methods can be found in books like Silverman [1986], Scott [1992]. When data dimension is high, these regular methods as well as density estimation itself suffer the so called "curse of dimensionality". It is also impossible to visualize a density surface when the dimension is higher than five.

It has been realized that in practical situations, the dimension of a true data structure is often much lower than the number of variables in the study. (Scott [1992]). Therefore it is desirable to project the high-dimension data onto an manageable lower dimensional subspace. In practice, projection pursuit method (Friedman, Stuetzle, and Schroeder [1984]) and some regular techniques like principal components decomposition are used to reduce the dimensionality.

Complex environmental modeling studies (Spear, Hornberger [1980]) result in many multivariate data analysis problems which are essentially density estimation problems. Many variables involved behave like noises. In fact, it seems that different sets of variables feature local data structures at different subspaces, while adding or removing other variables in these subspaces has little effects. If we can locate these subspaces and can identify the noises, the dimensionality will be greatly reduced.

In Section 2, we will discuss about the noises in density structure. In Section 3, the CART (Classification And Regression Tree) tree methodology (Breiman et al. [1985]) will be applied to construct a density estimation method according to the insights obtained from Section 2. Although the idea has been circulated among the authors of CART and other researchers, there are many serious problems in applying CART to density estimation. We solved these problems through defining a roughness parameter and following a tree optimizing approach developed by Shang [1993], Breiman and Shang [1994]. Similar type of application has been explored in contingency table analysis. (see Shang [1993], [1994]).

The performances of the method are studied through simulations. Spear, Grieb and Shang [1994] applied it to study the uncertainty of complex environmental modeling. Some techniques to enhance its interpretations were discussed in that paper.

2 Noises in Density Structure

In this section, we will provide an application background for multivariate density estimation. This will give us some insights about noises in density structure.

2.1 Pass Region and Density

An environmental model is usually very complex and is applicable to many similar environmental processes. When it is applied to a specific situation, it is necessary to study the sensitivity of the model to the local phenomena. Sensitivity analysis can help us to understand more about the model structure, to be more efficiently monitoring an environmental process and to build a more reliable procedure for risk assessment. A model can be simplified as:

$$Y = f(X, \theta) \tag{1}$$

where X is background information of a local experiment; Y is output; θ is the vector of parameters which varies in a parameter space Θ. At a local point X, we may be interest in when the model produces outputs similar to our observations or we may be interest in when the outputs exceed some extreme limits. We use C_Y to denote the region of these outputs and call it as a *criterion region*. The parameter set which produces outputs in C_Y is called a *pass region*:

$$\mathcal{P} = \{\theta : f(X, \theta) \in C_Y\} \tag{2}$$

The pass region summarizes information of parameter sensitivity with respect to the local background and the criterion.

It is usually impossible to solve equation (2) analytically. An alternative is to use Monte Carlo simulations. If we take a uniform sample from the parameter space Θ, then the points that produces outputs falling in C_Y construct a uniform sample from the pass region $\mathcal{P}$. Our problem is: how to reconstruct the pass region from the random sample. This is equivalent to estimate the indicator function of $\mathcal{P}$ or its smoothed version:

$$g(\theta) = \lim_{V_\theta \to 0} \frac{U_\theta}{V_0 V_\theta} \quad \theta \in \Theta \tag{3}$$

if the boundary of $\mathcal{P}$ is continuous. Here V_θ is the volume of N_θ, a small neighborhood of θ. U_θ is the volume of $N_\theta \bigcap \mathcal{P}$. V_0 is the volume of $\mathcal{P}$.

Notice that $g(\theta)$ is a density function. The process of getting a pass point can be defined as a $\Theta \to \Theta$ random variable ξ with $g(\theta)$ as its density. If there is a prior distribution $\pi(\theta)$ on Θ, then the corresponding distribution of parameters in $\mathcal{P}$ is just the posterior distribution $f(\theta \mid \xi)$. Now it is this posterior distribution rather than the region itself featuring the parameter sensitivity. If we take points from Θ according to $\pi(\theta)$, then the pass points construct a sample from $f(\theta \mid \xi)$.

In any case, the problem of parameter sensitivity analysis is essentially a density estimation problem. We are trying to recover some main features of a distribution from samples.

We may replace the terms *criterion region* by *critical region*, and *pass region* by *confidence region* in above context. Then we are dealing with a typical statistics problem: exploring features of a complex confidence region. Obviously, same idea can be applied to many other problems either in statistics or in other fields.

2.2 Noises and Dimensionality

The number of parameters (variables) in model (1) is usually very large. However it is expected that only a few of them will be useful in a local experiment. The traditional sensitivity analysis reduces the dimensionality through examining each individual variable. The simplest way is to compare the sample's range with the variable's range. The larger the difference, the more sensitive the variable.

This simplest way ignores the distribution of variables and their interactions. However if there is no interactions among variable and all variables are uniformly distributed, or equivalently if the variables as a whole follow a uniform distribution on a hyper-rectangle, the rectangle will completely feature the local experiment. Individually, if a variable is independent with the others and follows a uniform distribution in the variable's range, then the variable will be useless in featuring the local experiment. It behaves like a noise. We call it as a *global noise*.

It may not be easy to detect a global noise before estimating the density. There may not be enough global noises to reduce the dimensionality to a manageable level too. However, a variable may be important in some subspaces and completely have no influence in others. We call it as a *local noise*. Being able to locate these subspaces and to identify the local noises in each of the subspaces will greatly reduce the dimensionality of our problem.

Notice that if the underlying density is a smooth function and if one subspace is sufficiently small, the real density in the subspace can be approximated by a constant. In this subspace all variables can be considered as local noises. Therefore if we can find a way to partition the whole space into some subspaces such that the density is a constant (approximately) in each of the subspaces, then all features of the density will be summarized (approximatedly) by the partition process.

Here comes our approach of estimating high-dimension density: through identifying feature variables and noise variables, we partition the data space with the feature variables until all variables are local noises. The density can be estimated immediately by the constants in the subspaces and the density structure is summarized by the partition. The approach needs to be insensitive to noises. It is also necessary to organize the partition into a simple, understandable format.

3 Tree-Structured Density Estimation

The CART tree methodology has the features we desired at the end of last section. Some of its important ideas will be briefed in subsection 3.1. A few serious problems exist if we apply the method to density estimation directly. We will discuss these problems and their solutions in subsection 3.2 and 3.3. Simulation results are presented in subsection 3.4.

3.1 Tree-Structured Methodology

The procedure of applying the tree-structured methodology to a statistics problem includes the following steps: defining problem; defining splits; tree growing; pruning; and tree selection.

First we need to have a measure of lack of accuracy. It will decrease as the partition gets finer. We also need to define a pool of splits. The splitting rule decides which split should be selected from the pool. The tree will keep on growing until some stopping rules are reached. This results in a big tree.

The essential idea in CART is in the pruning and tree selection process. It adapts an idea from variable selection in regression analysis. The idea selects the best dimensionality rather than the best combination of variables as the first is more stable than the second from sample to sample. In CART, the dimensionality is the tree size (number of terminal nodes). The pruning algorithm produces a "best" subtree for each given tree size. This "best" subtree has the smallest estimation error among all possible subtrees with the same tree size. After the "best" tree list is established, an independent test data set or cross-validation will be applied to select the "best" tree size. Then the method produces the final "best" tree.

A tree optimizing approach was developed by Shang [1993], Breiman and Shang [1994] to make the "best" trees list even better. It tries to adjust the existing splits to nullify the effects made by the "greedy" nature of CART stepwise procedure.

3.2 Roughness Parameter

For density estimation, a measure of accuracy is the *mean integrated square error*:

$$MISE = E\int (f-\hat{f})^2 \tag{4}$$

Without loss of generality, we assume the data is defined on the p-dimension unit cubic: $S = (0,1)^p$. n random points $X_1, X_2, \ldots, X_n$ have been sampled from some underlying density f defined on S.

Suppose S has been partitioned into m subspaces:

$$S = \bigcup_{i=1}^{m} S_i \tag{5}$$

by some partition τ. Each S_i has volume V_i. τ also divides the n data points into m parts. Let n_i be the number of data points in S_i. Then the density can be estimated by:

$$\hat{f}(x) = d_i = \frac{n_i/n}{V_i} \quad \text{if } x \in S_i \tag{6}$$

For this estimator $\hat{f}$, the last two items of the integrated square error $\int(\hat{f}-f)^2$ can be calculated easily:

$$I(\tau) = \int \hat{f}^2 - 2\int \hat{f} f = -\sum_{i=1}^{m} \frac{(n_i/n)^2}{V_i} \tag{7}$$

If one of the V_i is very small, yet S_i still contains at least one data point, then $I(\tau)$ will be very small. If we use number of subspaces to build the "best" tree list, then rough estimations will be selected no matter what the real density is. A better smoothness measure should consider both the fineness (number of subspaces) and the evenness together.

We define a *roughness parameter* as the harmonic average of the volumes of the subspaces:

$$R(\tau) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{V_i} \tag{8}$$

It is easy to show that: $R(\tau) \geq m$ and $R(\tau) = m$ if and only if τ is the even partition, i.e. $V_i = 1/m$. In general, the larger the m and/or the more uneven of the τ, the larger the $R(\tau)$. These are the properties we desired. We also call $P(\tau) = m * R(\tau)$ as the *penalty function* of τ.

Since $R(\tau)$ takes continuous values, it is necessary to make it discrete. For a positive number q, we define:

$$C(k) = \{\tau : k-0.5 \leq qR(\tau) < k+0.5\} \quad k = 1, 2, \ldots \tag{9}$$

All τ in $C(k)$ will be considered to have same roughness parameter k. Here q is a scaling factor, normally we take it as 1.

Another possible *roughness* measure is:

$$R^1(\tau) = m^2\sum_{i=1}^{m} V_i^2 = \sum_{i=1}^{m}\left(\frac{V_i}{1/m}\right)^2 \tag{10}$$

It has better probability interpretations but penalizes unevenness less than $R(\tau)$ does.

3.3 Tree Growing and Pruning

Due to the restriction on paper length, here we will only discuss three issues which is more essential to density estimation.

Pool of Splits

Since the volumes of subspaces influence $I(\tau)$ too, there are essentially infinite ways to make a split. Our splits will be selected from the following:

$$\Sigma = \{\sigma : \sigma = (x_i \leq j/G) \quad 1 \leq i \leq p; 1 \leq j < G\} \qquad (11)$$

Here G is an positive integer. We usually take it as 100.

Splitting Rules

Suppose current subspace is S_0 and $\sigma \in \Sigma$ split S_0 into S_l and S_r. Let n_0, n_l, n_r and V_0, V_l, V_r be the corresponding number of data points and the volumes. Then the integrated square error defined in (7) will be decreased by:

$$\Delta I(\sigma) = \frac{n_0^2}{n^2} \frac{V_0^2}{V_0 V_l V_r} (\frac{V_l}{V_0} - \frac{n_l}{n_0})^2 \qquad (12)$$

Maximizing $\Delta I(\sigma)$ favors splits close to the boundaries of S_0, thus makes the estimation rough. At the same time, the penalty function P increases:

$$\Delta P(\sigma) = \frac{V_0^2 - V_l V_r}{V_0 V_l V_r} \qquad (13)$$

a better splitting rule would be: $\Delta I(\sigma)/\Delta P(\sigma)$. We actually use a simpler version:

$$D(\sigma) = (\frac{V_l}{V_0} - \frac{n_l}{n_0})^2 \qquad (14)$$

Maximizing $D(\sigma)$ is maximizing the distance between the empirical c.d.f of samples in subspace S_0 and the c.d.f of the local uniform density. As we are going to apply the tree optimizing procedures, this initial selection is not quite essential.

Tree Pruning

The roughness parameter $R(\tau)$ defined in (8) will be used to make the "best" tree list. For each positive integer k, we will try to find the subtree τ_k with smallest $I(\tau)$ among all subtrees in $C(k)$. A better definition of the "best" is:

$$\frac{I(\tau_k)}{R(\tau_k)} = \min_{\tau \in C(k)} \frac{I(\tau)}{R(\tau)} \qquad (15)$$

C_k is defined as in (9).

The roughness parameter $R(\tau)$ depends on the number of terminal nodes. It is impossible to generate the subtree in (15) directly. We make the list through a two-step approach. First, we build the "best" tree list according to the penalty function $P(\tau)$ although still using the criterion in (15). Then the "best" tree list for $R(\tau)$ can be obtained easily. Here $P(\tau)$ needs to be made discrete as in (9).

4 Simulations

Extensive simulations have been made to study the performance of the tree method. Two aspects need to be examined. The first, how the method captures the features. The second, how it responses to the influences of noises. Both contributes to the accuracy of estimation. In this paper, only the results of second aspect will be presented due to the limitation of paper length.

Three types of simulations are designed. Each of the underlying densities is a random composition of four simpler densities: $f(x) = \sum_{i=1}^{4} c_i g_i(x)$. Here (c_1, c_2, c_3, c_4) is a random point from the four dimension simplex. $f(x)$ is restricted in the unit hyper-cubic $(0,1)^p$.

For the first type of simulation, each $g_i(x)$ is a five dimensional normal densities: $N(\mu^i, \Sigma^i)$. μ^i comes from uniform distribution $U(1/6, 5/6)^5$. Σ^i is a diagonal matrix, the diagonal elements are from $U(0, 5/9)$. For each data set generated from the first type of simulation, five global noises are added to make the $g_i(x)$ of the the second type of simulation. The noises are from $U(0,1)^5$.

We consider local noises in the third type of simulation. Each of the $g_i(x)$ contains five feature variables and five noise variables. The noises are from $U(0,1)^5$ and the features are from a five dimensional normal densities as in the first type of simulation. Whether a variable is noise or not is randomly decided.

Each type of simulation is repeated eight times. As there are so many random factors in the design, they are essentially 24 different simulations. The sample size is 500. A Monte Carlo sample with 3,000 points is used to calculate the integrated square loss $\int(f - \hat{f})^2$. The results are compared with results from kernel estimation. In the kernel estimation, different window sizes are used for different variables. The best window sizes are selected through a grid search. The window sizes change from 0 to 1 at a step of 0.02. The simulation results are presented in Table 1.

In Design 1, there is no designed noises. So Both methods are trying to capture the features. Although the tree estimation uses a jumped step function to estimate a smooth function, its accuracy is comparable with kernel's. In two of the situations, it is even better. However when global noises are added in Design 2, the efficiency of kernel estimation reduces dramatically

Simulation	*Design 1*		*Design 2*		*Design 3*	
	kernel	tree	kernel	tree	kernel	tree
1	1.190	1.305	3.145	1.309	2.830	0.807
2	6.616	5.986	19.000	6.102	5.086	3.050
3	0.628	0.816	1.800	0.816	1.864	0.852
4	0.869	1.305	2.328	1.343	1.635	0.782
5	0.927	0.867	2.768	0.886	1.829	1.082
6	1.080	2.195	3.969	2.195	1.133	0.631
7	4.797	7.028	14.125	7.475	1.928	1.132
8	0.633	0.715	3.545	0.830	2.014	1.115

Table 1: Comparing Tree-based Estimation and Kernel Estimation

while there is hardly any changes in tree estimation. In Design 3, a variable could be a feature here and a noise there. Still tree estimation is much better.

5 Summary and Conclusions

The information carried by a density function is how the density is distributed or how it is different from a uniform distribution. If a variable is uniformly distributed and it is independent with the others, it should be considered as a noise. In a high-dimension situation, many variables may behave like local noises: they are conditionally independent with other variables and are uniformly distributed in some subspaces. A good density estimation method should be able to identify these noises while capturing the real data features.

The tree-structured density estimation has the power to identify these noises locally. The method is based on the CART tree methodology. However significant changes have been made to adapt the methodology into density estimation. These include defining a roughness parameter and applying a tree optimizing approach. The method is compared with kernel method through simulations. The simulations presented in this paper concern only the influence of noises.

More simulations are made to study how the method captures the features of density functions in one or two dimension situations. Although the underlying densities are continuous, the tree method still have comparable performance when it is compared with other regular methods. Actually it is often more accurate when the underlying density is complex. These results will be presented in a more complete paper.

Acknowledgements

This work is supported by the Chevron Postdoctorate Fellowship. The author would like to thank Professor Spear for his constant support and encouragements. His comments and insights from application perspective are very enlightening and valuable.

References

L. Breiman, J. H. Friedman, A. Olshen, C. J. Stone, *CART: Classification and Regression Trees,* Wadsworth, Belmont, 1984.

L. Breiman, N. Shang, "Optimizing Trees," *In preparation,* 1994.

J. H. Friedman, W. Stuetzle, A. Schroeder, "Projection Pursuit Density Estimation," *J. Amer. Statist. Assoc.*, Vol. 79, pp. 599-608, 1984.

D. V. Scott, *Multivariate Density Estimation,* John Wiley and Sons, 1992.

N. Shang, "New Developments in Tree-Structured Methodology," *Ph.D Dissertation*, University of California, Berkeley, 1993.

N. Shang, "Local Models In Big Contingency Table Analysis.," *In preparation* , 1994.

B. W. Silverman, *Density Estimation,* Chapman and Hall, 1986.

R. C. Spear, T. M. Grieb, N. Shang "Parameter Uncertainty and Interaction in Complex Environmental Models," *Water Resources Research*, accepted. 1994.

R. C. Spear, G. M. Hornberger, "Eutrophication in Peel Inlet: II. Identification of Critical Uncertainties Via Generalized Sensitivity Analysis," *Water Research*, Vol. 14, pp. 43-49, 1980.

From Hypercubes to Permutation Polytopes: A Geometric Analysis of Paired Comparisons

Keith Baggerly, Los Alamos National Laboratory
Statistics Group, P.O. Box 1663 (MS F600)
Los Alamos, NM 87545

Abstract: Rankings of n items are often constructed from paired comparisons within the set (football rankings, for example). Collections of paired comparisons, however, are subject to inconsistencies: *e.g.*, $A > B$, $B > C$, $C > A$. Treating the set of all possible outcomes of the $\binom{n}{2}$ paired comparisons of n items as vertices on a hypercube, inconsistencies can be removed by orthogonal decomposition. The resulting consistent substructures lie imbedded in permutation polytopes.

1. Introduction

Given a set of n items, it is often desirable to be able to assign a ranking to the set of items, indicating first place, second place, and so on. A situation that often arises is that the ***rankings*** of n items are not given *per se*; rather, the data consist of the outcomes from some subset of the $\binom{n}{2}$ possible paired comparisons, from which a ranking must be inferred. For example, if the three items A, B, C were to be compared in pairs, the collection of outcomes $A > B$, $A > C$ and $B > C$ would imply that item A should be ranked first, item B second, and item C third (here and in the sequel, $A > B$ means that item A is preferred to item B).

A common problem that arises with paired comparisons is that some collections of outcomes are internally inconsistent, *e.g.* $A > B$, $B > C$, and $C > A$. These inconsistent triples were denoted *circular triads* by Kendall and Babington Smith (1940). How to deal with these inconsistent groupings is a matter of some dispute. In most formulations of ranking models, inconsistent collections of outcomes are simply ignored. However, some information can occasionally be gleaned from inconsistent collections. Given four items and the collection of outcomes $A > B$, $A > C$, $A > D$ and $B > C$, $C > D$, $D > B$, it can be asserted that item A is preferred to the other three, even though the relative ordering amongst items B, C and D is not discernible.

2. Permutation Polytopes as Projections of Hypercubes

When a comparison of two items is conducted, the outcome is binary – either item A is preferred to item B, or item B is preferred to item A. Each paired comparison can thus be thought of geometrically as defining an axis, for example the AB axis, where the particular outcome determines the value along that axis: 1 if (using the example above) A is preferred to B, and -1 ($-AB$) if B is preferred to A. In this manner, a space containing the overall structure arising from a collection of paired comparisons can be specified by the cartesian product of these axes. The collections of outcomes of the $\binom{n}{2}$ paired comparisons arising from the possible pairings of 2 out of n items can be viewed as vertices on an $\binom{n}{2}$-dimensional hypercube, with coordinates of either 1 or -1. These vertices can then be thought of as points in $\Re^{\binom{n}{2}}$. As an example, consider the collections of paired comparisons possible among three items, A, B, C. If (for purposes of orientation) the axes defined are taken to correspond to AB, AC and BC, respectively, then $(1,1,1)$ would indicate $A > B$, $A > C$, and $B > C$, $(-1,-1,1)$ would indicate $A < B$, $A < C$ and $B > C$, and so on. The eight possible collections correspond to the vertices of a cube in three dimensions.

To address the problem of inconsistent sets of comparisons, consider the cube defined by the collections of paired comparisons of three items. Of the eight vertices, two correspond to triples which are linearly inconsistent: $(1,-1,1)$ and $(-1,1,-1)$, using the AB-AC-BC coordinate system as before. The other six vertices correspond to the six possible rankings of three items. The two inconsistent triples both lie along a single vector through the origin, $\overrightarrow{ABC} = (1,-1,1)$. (In the sequel, the notation $\overrightarrow{IJK}$ will be used to indicate the vector corresponding to an inconsistent arrangement of the three arbitrary items I, J, K, specifically $I > J$, $J > K$ and $K > I$). This vector defines an "inconsistent subspace" associated with this set of paired comparisons; this in turn suggests the existence of a "consistent subspace". The linear (ranking) information present within a collection of paired comparisons can be viewed as a function of the projection of the vector associated with that particular vertex of the hypercube onto the consistent subspace. This projection is illustrated in Figure 1.

To establish the general procedure, we need to show that a decomposition into inconsistent and consistent subspaces is always feasible.

As a first step, note that any triplet of items can give rise to inconsistent pairings. There are $\binom{n}{3}$ item triplets, defining an equal number of vectors corresponding to inconsistencies. These vectors must be linearly dependent, as $\binom{n}{3}$ grows faster than $\binom{n}{2}$ (the dimension of the hypercube). Thus, it is necessary to establish the dimension

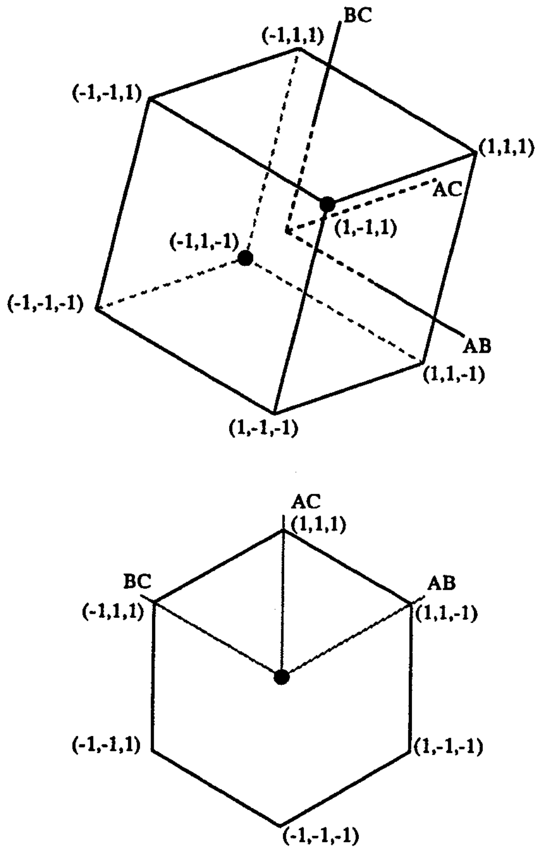


Figure 1: Cube of Paired Comparisons of Three Items, and the Associated Projection onto the Consistent Subspace. Inconsistent Triples are Indicated by Dark Circles. Coordinates are in the (AB, AC, BC) System.

Table 1: Vectors Associated with Inconsistent Triples Arising from Paired Comparisons of Four Items.

		Axis Labels					
		AB	AC	BC	AD	BD	CD
Inconsistent	$\overrightarrow{ABC}$	1	-1	1	0	0	0
Triples:	$\overrightarrow{ABD}$	1	0	0	-1	1	0
$\overrightarrow{ABC}$ points to	$\overrightarrow{ACD}$	0	1	0	-1	0	1
A>B,B>C,C>A	$\overrightarrow{BCD}$	0	0	1	0	-1	1

of the space spanned by the vectors corresponding to these inconsistent triples.

Consider the vectors corresponding to the inconsistent triples arising from the comparisons of four items. These are shown in the rows of Table 1. An entry of 1 in the table indicates that the specified pair was preferred in the given order, a -1 indicates that the pair was preferred in the reverse order, and a 0 indicates that no direct pairing has occurred. As $\overrightarrow{BCD} = \overrightarrow{ABC} - \overrightarrow{ABD} + \overrightarrow{ACD}$, the vectors are linearly dependent. However, if attention is constrained solely to the triples containing a specific item (*e.g.* A), those vectors are not linearly dependent.

Lemma 1: The vectors corresponding to inconsistent triples involving item A are linearly independent.

Proof: For any items I and J, the vector $\overrightarrow{AIJ}$ has a 1 in the entry corresponding to the IJ axis, and the vectors corresponding to all other inconsistent triples containing A have a 0. Thus, $\overrightarrow{AIJ}$ cannot be formed as a linear combination of such vectors.

Lemma 2: Any vector corresponding to an inconsistent triple can be expressed as a linear combination of vectors corresponding to inconsistent triples involving item A.

Proof: As the lemma is trivially true if the inconsistent triple involves A, it suffices to show that it holds for an arbitrary inconsistent triple I, J, K not involving A. This is most easily shown using unit vectors corresponding to the various paired comparison axes; *e.g.*, $\overrightarrow{ABC} = \vec{e}_{AB} - \vec{e}_{AC} + \vec{e}_{BC}$.

$$\begin{aligned}
&\overrightarrow{IJK} \\
&= \vec{e}_{IJ} - \vec{e}_{IK} + \vec{e}_{JK} \\
&= \vec{e}_{IJ} - \vec{e}_{IK} + \vec{e}_{JK} + \\
&\qquad (\vec{e}_{AI} - \vec{e}_{AI}) + (\vec{e}_{AJ} - \vec{e}_{AJ}) + (\vec{e}_{AK} - \vec{e}_{AK}) \\
&= (\vec{e}_{AI} - \vec{e}_{AJ} + \vec{e}_{IJ}) - (\vec{e}_{AI} - \vec{e}_{AK} + \vec{e}_{IK}) + \\
&\qquad (\vec{e}_{AJ} - \vec{e}_{AK} + \vec{e}_{JK}) \\
&= \overrightarrow{AIJ} - \overrightarrow{AIK} + \overrightarrow{AJK}.
\end{aligned}$$

Lemma 3: Any vector corresponding to an inconsistent k-tuple (for example, $\overrightarrow{ABCD} = \vec{e}_{AB} + \vec{e}_{BC} + \vec{e}_{CD} - \vec{e}_{AD}$ corresponds to an inconsistent 4-tuple) can be written as a linear combination of vectors corresponding to inconsistent triples.

Proof: The lemma holds trivially if $k = 3$, and inconsistency is impossible if $k < 3$, so the lemma holds then as well. If $k > 3$, the vector corresponding to the inconsistent k-tuple can be written as the sum of a vector corresponding to an inconsistent triple and a vector corresponding to an inconsistent $(k-1)$-tuple as follows:

$$\begin{aligned}
&\overrightarrow{IJKL\ldots N} \\
&= \vec{e}_{IJ} + \vec{e}_{JK} + \vec{e}_{KL} + \ldots - \vec{e}_{IN} \\
&= (\vec{e}_{IK} - \vec{e}_{IK}) + \vec{e}_{IJ} + \vec{e}_{JK} + \vec{e}_{KL} + \ldots - \vec{e}_{IN} \\
&= (\vec{e}_{IJ} - \vec{e}_{IK} + \vec{e}_{JK}) + \vec{e}_{IK} + \vec{e}_{KL} + \ldots - \vec{e}_{IN} \\
&= \overrightarrow{IJK} + \overrightarrow{IKL\ldots N}.
\end{aligned}$$

The $(k-1)$-tuple can then be reduced, and the lemma follows by induction.

Theorem 1: Given the space defined by the $\binom{n}{2}$ axes associated with the paired comparisons possible among n items, the set of vectors corresponding to inconsistent triples involving item A forms a basis for the inconsistent subspace.

Proof: The theorem follows immediately from Lemmas 1-3.

Corollary 1.1: The dimension of the inconsistent subspace is $\binom{n-1}{2}$.

Corollary 1.2: The consistent subspace exists and has dimension $\binom{n}{2} - \binom{n-1}{2} = n - 1$.

Further, it can be shown (cf. Baggerly (1994)) that if $\overrightarrow{I1}$ is the vector corresponding to item I being ranked first (item I beats all other items, and pairings of items not including I do not occur) then the collection of vectors $\{(\overrightarrow{I1} + \alpha\overrightarrow{A1})/\sqrt{n}\}$, where I is any item other than A and $\alpha = \frac{1+\sqrt{n}}{n-1}$, forms an orthonormal basis for the consistent subspace.

Putting this basis into use, the projection of the six-dimensional hypercube arising from the pairwise comparisons of 4 distinct items onto the corresponding 3-dimensional consistent subspace is shown in Figure 2. The completely consistent sets of paired comparisons correspond to rankings of the four items; these are situated at the vertices of the resultant polytope. In terms of cartesian coordinates, these vertices lie at the 24 permutations of $(0, \pm 1, \pm 2)$. Those sets of paired comparisons with only one inconsistent triple (slightly inconsistent) are situated at the centers of the hexagonal faces of the resultant polytope; these each have multiplicity 2 (*i.e.*, two vertices of the initial hypercube map to each such point). In terms of cartesian coordinates, these vertices lie at the 8 permutations of $(\pm 1, \pm 1, \pm 1)$. Finally, those sets of paired comparisons with two inconsistent triples (grossly inconsistent) are situated behind the square faces of the resultant polytope; these each have multiplicity 4. In terms of cartesian coordinates, these vertices lie at the 6 permutations of $(0, 0, \pm 1)$. It is impossible to have more than two inconsistent triples arising in the paired comparisons of 4 items. The edges defining the convex hull of these points are also shown.

Several features of this figure should be noted. First, each edge of the initial hypercube has been shortened by the same amount. This equal shortening follows from the fact that each edge of the hypercube corresponds to a shift along a single paired comparison axis and the projection acts upon the axes in a symmetric manner. Thus, the vertex labelled *CBAD* is the same distance from the circled points as it is from the vertex labelled *CBDA*. Second, the projections of the hypercube vertices corresponding to full rankings of the items define the convex hull of the projection of the hypercube onto the consistent subspace. The polytope thus defined is a truncated octahedron, having eight hexagonal faces and six square faces, and is equivalent to the permutation polytope associated with the rankings of four items.

3. Permutation Polytopes

A permutation polytope is the convex hull defined by the $n!$ points in $\Re^n$ whose coordinates are permutations of the first n integers. Using these polytopes in to analyze ranked data was first suggested by Schulman (1979); this procedure has recently been generalized and expanded on by Thompson (1993).

Consider the rankings of four items, A, B, C, D, and let $\vec{\pi}_i$ be a vector in $\Re^4$ whose coordinates are the ranks of A, B, C, D, respectively. Thus, $\vec{\pi}_i = (1, 2, 3, 4)$ would correspond to the ordering $\langle A, B, C, D \rangle$; item A is ranked first, item B second, item C third, and item D fourth. Similarly, $\vec{\pi}_i = (3, 4, 1, 2)$ would correspond to the ordering $\langle C, D, A, B \rangle$; item A is ranked third, item B fourth, item C first, and item D second.

As each vector $\vec{\pi}_i$ has the same components,

$$\sum_{j=1}^{4} \vec{\pi}_i(j) = 1 + 2 + 3 + 4 = 10$$

a constant, so this polytope is constrained to lie in a 3-dimensional subspace of $\Re^4$. Similarly, as the average of the components, $\bar{\pi}_i$, is always the same,

$$\sum_{j=1}^{4} (\vec{\pi}_i(j) - \bar{\pi}_i)^2 = 2.25 + .25 + .25 + 2.25 = 5,$$

another constant, so this polytope is constrained to lie on a 4-dimensional hypersphere. These constraints generalize to n dimensions, so a permutation polytope must lie imbedded in an $(n-1)$-dimensional hypersphere.

Two vertices of the permutation polytope are joined by an edge if and only if they differ by a single transposition of two consecutive integers: $(3, 4, 1, 2)$ is joined to $(3, 4, 2, 1)$, $(2, 4, 1, 3)$ and $(4, 3, 1, 2)$. In terms of item orderings, this transposition of consecutive integers corresponds to a transposition of two adjacent items: $\langle C, D, A, B \rangle$ is joined to $\langle D, C, A, B \rangle$, $\langle C, A, D, B \rangle$, and $\langle C, D, B, A \rangle$.

The connection structure imparts a powerful property to the permutation polytopes: every face on the polytope has a direct interpretation in terms of rankings.

This feature is illustrated in Figure 2, where eight of the vertices have been labelled with their corresponding

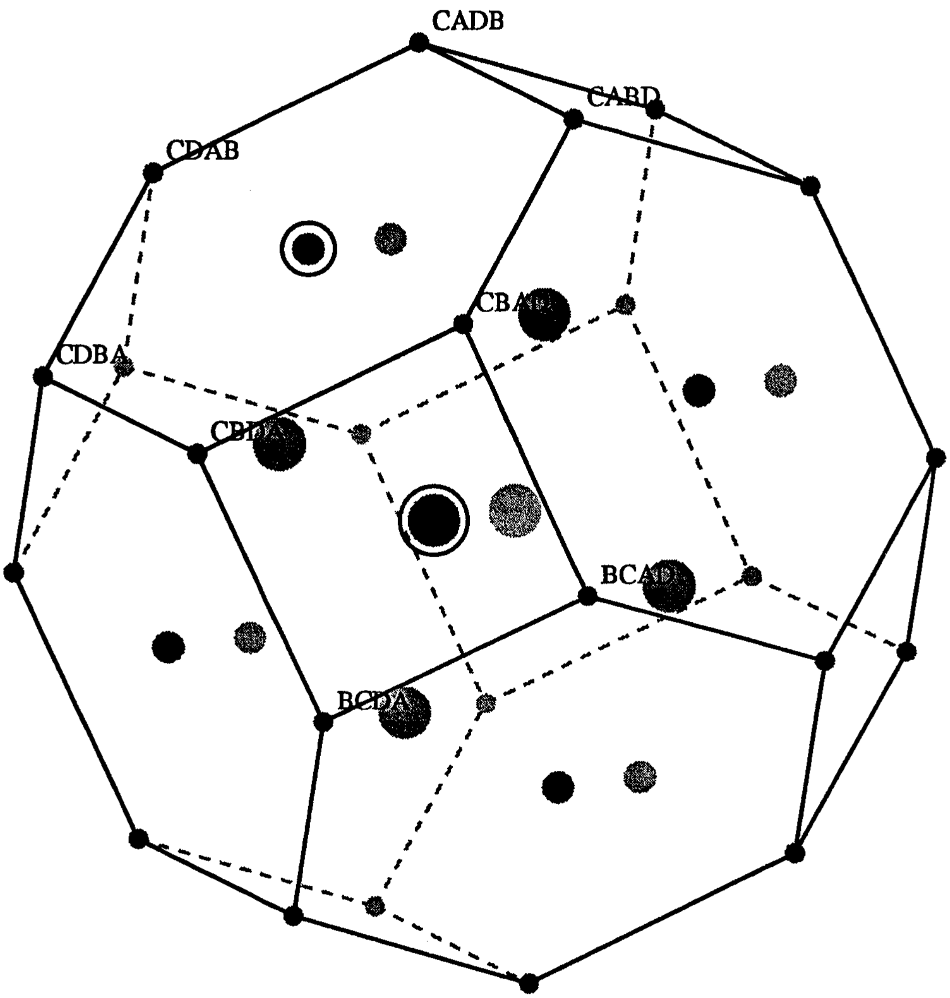


Figure 2: Projection of the 6-dimensional Hypercube Arising from the Paired Comparisons of Four Items onto the Consistent Subspace.

item orderings. The labelled hexagon in the upper left corresponds to the collection of all rankings in which item C is ranked first. Similarly, three of the remaining hexagonal faces correspond to items A, B and D being ranked first, respectively. The other four hexagonal faces correspond to a given item being ranked last. The six square faces correspond to a given pair of items being ranked in the first two positions; the labelled square in front corresponds to items B and C being jointly ranked first and second. Methods of determining what faces can arise and assigning interpretations to them are provided in Thompson (1993) and Baggerly (1994).

Hence, the ranking information present in a collection of paired comparisons can be inferred by noting what face of the permutation polytope is indicated by the projection of the collection onto the consistent subspace. The inconsistent collections mapping to the circled dot at the center of the hexagonal face in the upper left of Figure 2 correspond to item C being ranked first, while the relative ranking of items A, B and D is left indeterminate. Similarly, the large circled dot behind the square face in the center indicates a slight preference for items B and C over items A and D. The fact that the vertex is within the convex hull (as opposed to on the surface) indicates that some ambiguity is present. In general, the magnitude of the projection onto the consistent subspace can be taken as an indicator of the strength of the expressed preference.

4. Future Work

Several avenues remain to be explored. There are questions of what to do if the collections of paired comparisons are incomplete (in that not every comparison has been made), or if some comparisons have been made multiple times. It is not immediately clear how to scale the projections to account for the missing information. These problems have been addressed analytically by Kendall (1955) and recently by Andrews and David (1990). A good overview of much of the analytic work done on paired comparisons is provided by David (1988).

If all comparisons have been made, but some ties have resulted (yielding a 0 as opposed to a 1 or -1 in the appropriate entry), the projection onto the consistent subspace is still well-defined. Every full ranking can be represented as a collection of paired comparisons; if ties are allowed, every partial ranking (*e.g.*, A first) can also be represented by a collection. This fact suggests new geometric ways of examining mixtures of full and partially ranked data.

Finally, there may exist other projections of the paired comparison hypercubes which may be of interest, as these other projections can potentially reveal trends in exactly how inconsistencies tend to occur.

References

Andrews, D. M. and David, H. A. (1990) "Nonparametric Analysis of Unbalanced Paired-Comparison or Ranked Data", *Journal of the American Statistical Association*, **85**, 1140-1146.

Baggerly, K. A. (1994), *Visual Estimation of Structure in Ranked Data*, Ph.D. Thesis, Department of Statistics, Rice University.

David, H. A. (1988), *The Method of Paired Comparisons*, Charles Griffin & Company, Ltd., London; Oxford University Press, New York.

Kendall, M. G. (1955), "Further Contributions to the Theory of Paired Comparisons", *Biometrics*, **11**, 43-62.

Kendall, M. G., and Babington Smith, B. (1940), "On the Method of Paired Comparisons", *Biometrika*, **31**, 324-345.

Schulman, R. S. (1979), "A Geometric Model of Rank Correlation", *The American Statistician*, **33**, 77-80.

Thompson, G. L. (1993), "Generalized Permutation Polytopes and Exploratory Graphical Methods for Ranked Data", *Annals of Statistics*, **21**, 1401-1430.

On the Analysis of Multiple Correlated Binary Endpoints in Medical Studies

Chung-Kuei Chang and Dror M. Rom

Department of Biostatistics, Rhône-Poulenc Rorer Central Research,
Collegeville, PA 19426, U.S.A.

ABSTRACT

A new procedure is proposed for the analysis of multiple correlated binary endpoints. The procedure is based on the exact distribution of $-2\Sigma_i \log(p_i)$, where p_i's are transformations of the statistics z_i's used to test the individual endpoints. We show how to make global, as well as local inferences regarding the hypotheses. We also compare this approach with several recently proposed multiple comparison procedures for the analysis of multiple correlated binary endpoints in terms of Type-I error control, and power.

Key words: endpoints, multiple comparison procedures, familywise error rate.

INTRODUCTION

Suppose there are c treatment groups, a control group and c-1 treated groups with increasing doses, and k endpoints with binary response were measured on each experimental unit. For endpoint i, i = 1, 2,..., k, we test the null hypothesis H_i of no treatment effect, against the alternative hypothesis that the response rate increases with dose. It is well known that the Type I error for testing $H_0 = \cap_i H_i$ could increase if we conclude that there is a treatment effect by rejecting H_0 when observing any significant result among the k endpoints. Several methods are available in the literature to control the overall Type I error. These include the Bonferroni procedure and its improvements, [see Holm (1979), Simes (1986), Hommel (1988), Rom (1990)], and procedures taking discreteness into account [see Brown and Fears (1981), Heyse and Rom (1988), Westfall and Young (1989), Tarone (1990), Rom (1992)]. In this paper, we study the conditional exact test of Fisher's combination procedure using $T = -2\Sigma_i \log(p_i)$ as test statistic, where p_i is the asymptotic p-value for testing the i-th endpoint.

NOTATION

Suppose that there is a total of r ($r \le 2^k$) different combinations of binary responses (response vectors) from the k endpoints, denoted by $\mathbf{D} = [\, d_{mi} \,]_{r \times k} = [\, \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_r]'$, where $d_{mi} = 1$, if the i-th endpoint in the m-th combination has response; else $d_{mi} = 0$; $m = 1, 2, \ldots, r$; $i = 1, 2, \ldots, k$. Let $\mathbf{n}_m' = [\, n_{m1}, n_{m2}, \ldots, n_{mc}]$ be the number of subjects in the c groups corresponding to the m-th response vector $\mathbf{d}_m'$, $m = 1,2,\ldots, r$, then our test procedure is based on $\mathbf{G} = [\, \mathbf{D}_{rxk} \mid \mathbf{N}_{0\ rxc}]$, where $\mathbf{N}_0 = [\, \mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_r]'$. Notice that the m-th row margin $n_{m\cdot} = \Sigma_{j=1}^{c} n_{mj}$ is the total number of subjects among the c groups that have response vector $\mathbf{d}_m'$, and the j-th column margin $n_{\cdot j} = \Sigma_{m=1}^{r} n_{mj}$ is the size of group j; $m = 1, 2, \ldots, r$; $j = 1, 2, \ldots, c$.

ANALYSIS OF INDIVIDUAL ENDPOINTS

For each endpoint i, a 2 x c table $\mathbf{E}_i = [\, e_{ilj}]$ can be derived from $\mathbf{G}$, where the cell count of the j-th column in the first (second) row is the number of subjects that do not respond (respond) for endpoint i, *i.e.*,

$$\begin{aligned} e_{ilj} &= \sum_{m=1}^{r} n_{mj} \mathbf{I}_{\{l-1\}}(d_{mi}), \\ \mathbf{I}_{\{l-1\}}(d_{mi}) &= 1, \quad \text{if } d_{mi} = l\text{-}1, \\ &= 0, \quad \text{otherwise;} \\ i = 1, 2, \ldots, k;\ l &= 1,2;\ j = 1, 2, \ldots, c. \end{aligned} \tag{1}$$

The analysis for endpoint i can be done by Mantel's score test using:

$$T_i = \sum_{l=1}^{2} \sum_{j=1}^{k} u_l \cdot v_j \cdot e_{ilj}, \text{ where } u_l = l - 1 \text{ and } v_j = j - 1. \tag{2}$$

The column scores v_j's reflect the progressive response at increasing doses. For other possible values of the scores, see Tukey, Ciminera and Heyse (1985). To test an upward trend in the response rate, the asymptotic p-value is:

$$p_i = 1 - \Phi(Z_i), \text{ where } Z_i = \frac{T_i - E(T_i)}{\sqrt{Var(T_i)}}\,. \tag{3}$$

OVERALL TEST PROCEDURE

To test the overall null hypothesis H_0 of no treatment effect at any endpoint, we propose to use:

$$T = -2 \sum_{i=1}^{k} \log(p_i), \tag{4}$$

which is equivalent to using:

$$T' = \prod_{i=1}^{k} p_i \, . \tag{5}$$

When the k endpoints are continuous and independent, T has chi-square distribution with $2k$ degrees of freedom, and it is known as Fisher's combination procedure. In our case, the overall p-value is calculated using the exact distribution of T', conditional on the row and column margins of the observed r x c table $\mathbf{N_0}$. It is the probability of observing any r x c table **N**, under the null hypothesis and conditional on the margins of $\mathbf{N_0}$, which is at least as extreme as $\mathbf{N_0}$, where the extremity is measured by T'. We denote the overall p-value by adj-p, and express it as:

$$\begin{aligned} \text{adj}-\text{p} &= \Pr(T' \leq T'_o | \text{margins of } \mathbf{N_0}) \\ &= \sum_{T' \leq T'_o} \Pr(\mathbf{N} | \text{margins of } \mathbf{N_0}) \, , \end{aligned} \tag{6}$$

where T'_o is the observed test statistic. Notice that the dependence of the k endpoints is reflected by the row margins and that N, conditional on the margins, follows multivariate hypergeometric distribution under the null hypothesis.

Our algorithm is as follows:

1. Calculate the observed statistic $T'_o = \prod_i p_i$, using **D** and the observed table $\mathbf{N_0}$.
2. Set adj-p to zero.
3. Enumerate r x c tables **N** satisfying the row and column margins of $\mathbf{N_0}$.
4. Use **D** and the enumerated table **N** to form k 2 x c tables and calculate their asymptotic p-values.
5. Calculate the corresponding test statistic $T' = \prod_i p_i$.
6. If T' is less than or equal to T'_o, then adj-p is increased by p, where p is the probability of observing the enumerated table N, conditional on the margins.
7. Return to 3, until all possible (given the margins) r x c tables are enumerated.

SOME COMPUTING ISSUES IN OUR PROGRAM

We have implemented our procedure in SAS using complete algorithm (see Verbeek and kroonenberg, 1985) to enumerate all the r x c tables satisfying the margins. When large problems are encountered, we use Monte Carlo simulation to estimate the adjusted p-value by taking random samples from all possible tables (see Boyett, 1979). Gail and Mantel's (1977) method to approximate the total number of r x c tables satisfying the margins can help making the decision of selecting the exact or Monte Carlo procedures.

EXAMPLE 1

The following example is from Rom (1992). In a carcinogenicity study, 100 mice were randomly assigned to either control or tested groups, with 50 mice in each group. Tumor incidence at site A and B were observed from each mouse. The experimental outcome is summarized in Table 1, where **D** is the 4 x 2 table under Endpoint 1 and Endpoint 2, and $\mathbf{N_0}$ is the 4 x 2 table under Group 1 and Group 2. From Table 1, we can derive two 2 x 2 tables for endpoint 1 and 2, and calculate the observed test statistic T'_o (see Table 2). Then, we start enumerating 4 x 2 tables **N** satisfying the margins of $\mathbf{N_0}$. From each enumerated table **N**, two 2 x 2 tables are derived and the corresponding test statistic T' is calculated. If $T' \leq T'_o$, the overall p-value adj-p is increased by the probability of observing **N**, under the null hypothesis and conditional on the margins of $\mathbf{N_0}$. Part of the SAS output is shown in Table 3. There are 128 tables satisfying the margins, and the overall p-value is about 0.0190.

Note that for this example, the Fisher's Exact test statistic for site A (B) is the number of mice in the treated group with tumor A (B), *i.e.*, 8 (5).

Table 1. Summary of number of mice with tumors

Tumor site	Endpoint 1 (Tumor A)	Endpoint 2 (Tumor B)	Group 1 (Control)	Group 2 (Treated)	Total
	D		$\mathbf{N_0}$		
None	0	0	48	39	87
A only	1	0	1	6	7
B only	0	1	0	3	3
A and B	1	1	1	2	3
Group size			50	50	100

Table 2: 2 x 2 tables for A and B generated from Table 1

Tumor	Incidence	Control	Treated	Asymptotic p-value
A	No	48	42	
	Yes	2	8	0.02275
B	No	49	45	
	Yes	1	5	0.04606

$T_0 = 0.02275 \text{ x } 0.04606 = 0.001048$

Table 3: Part of SAS output for Example 1

Obs	n_{11}	n_{21}	n_{31}	p_1	p_2	T'	p	adj-p	A	B
1	50	0	0	.0004	.0058	.0000	.00005	.00005	10	6
2	49	1	0	.0038	.0058	.0000	.00046	.00051	9	6
3	49	0	1	.0004	.0461	.0000	.00020	.00071	10	5
4	49	0	0	.0038	.0461	.0002	.00020	.00090	9	5
5	48	2	0	.0228	.0058	.0001	.00173	.00264	8	6
6	48	1	1	.0038	.0461	.0002	.00173	.00437	9	5
⋮										
127	38	6	3	.9962	.9942	.9904	.00046	.01900	1	0
128	37	7	3	.9996	.9942	.9938	.00005	.01900	0	0

$T'_0 = 0.001048$

Note:

1. p_1 is the asymptotic p-value for site A.
2. p_2 is the asymptotic p-value for site B.
3. p is the probability of observing N.
4. A is the number of tumors at site A in the treated group.
5. B is the number of tumors at site B in the treated group.

MAKING LOCAL INFERENCES

When rejecting the global null hypothesis H_0, we conclude that at least one of the H_i's is false. The closure principle of Marcus, Peritz and Gabriel (1976) can be employed to make inferences on individual hypotheses. With two endpoints, one can reject any individual hypothesis H_i, $i = 1,2$, if $H_0 = H_1 \cap H_2$ is rejected at level α, and H_i is also rejected at level α using the same procedure.

In our example, the hypothesis corresponding to tumor site A has a corresponding p-value of 0.0458 (see Table 4). Since both the global null hypothesis and this individual hypothesis are rejected at level 0.05, we can conclude that treatment causes an increase in the rate of tumor A. Note that our procedure, when applied to one endpoint only, is equivalent to Fisher's Exact test.

Table 4: Joint distribution and rejection regions

		B								
		0	1	2	3	4	5	6	Margin	
	0	.0001	.0002	.0003	.0001	.0000	.0000	.0000	.0006	
	1	.0005	.0019	.0028	.0017	.0003	.0000	.0000	.0072	
	2	.0017	.0080	.0136	.0106	.0036	.0004	.0000	.0380	
	3	.0035	.0182	.0366	.0353	.0164	.0031	.0001	.1131	
	4	.0040	.0250	.0600	.0698	.0409	.0110	.0009	.2114	
A	5	.0026	.0212	.0622	.0872	.0622	.0213	.0026	.2593	
	6	.0009	.0110	.0409	.0698	.0600	.0250	.0040	.2114	
	7	.0001	.0031	.0164	.0353	.0366	.0182	.0035	.1131	
	8	.0000	.0004	.0036	.0107	.0136	.0080	.0017	.0380	
	9	.0000	.0000	.0003	.0017	.0028	.0019	.0005	.0072	.0458
	10	.0000	.0000	.0000	.0001	.0003	.0002	.0001	.0006	
Margin		.0133	.0889	.2367	.3223	.2367	.0889	.0133		
							.1022			

Rejection region: Ordered p-values (Rom) ━━━━━━━━

Product of p-values ────────

Table 5: Actual levels and p-values

Method	Rejection Region	Actual level	P-value
Bonferroni	$P_{(1)} \leq 0.025$ $\equiv A \geq 9$ or $B = 6$	0.02050	0.0916
Heyse and Rom (1988)	$P_{(1)} \leq 0.0133$ $\equiv A \geq 9$ or $B = 6$	0.02050	0.0568
Rom (1992)	$\{P_{(1)} < 0.0458\} \cup$ $\{P_{(1)} = 0.0458$ & $P_{(2)} \leq 0.3389\}$	0.04226	0.0286
Proposed	$P_1 \cdot P_2 \leq 0.004306$	0.04664	0.0190

Note:

1. $P_{(1)}$ and $P_{(2)}$ are the ordered p-values of the individual endpoints, where $P_{(1)} \leq P_{(2)}$.
2. P_1 and P_2 are the asymptotic p-values.

COMPARISON WITH OTHER PROCEDURES

Using the above example, we compare our method with the Bonferroni procedure and two other exact procedures: Heyse and Rom (1988) procedure using the minimum p-value of the *k* endpoints as test statistic, and Rom (1992) procedure using the ordered p-values of the *k* endpoints as test statistic. The joint distribution of the tumor incidence at site A and B in the treated group, as well as the rejection regions of two exact procedures, are displayed in Table 4. The joint distribution can be obtained from Table 3 by summing up the hypergeometric probabilities of identical tumor incidence at the two sites in the treated group, where

(A, B) = (8, 5) is the observed value. The p-values and actual significance levels under $\alpha = 0.05$ are summarized in Table 5. We can see that the proposed procedure has the smallest p-value and the least conservative Type I error control ≤ 0.05.

CALCULATION OF POWER FUNCTION

Here we only discuss the case of two treatment groups with two endpoints. The method can analogously be extended to the general case. Assume P_{ij} is the actual response rate of the i-th endpoint at the j-th group and that V_j is the covariance between the two responses, where $i, j = 1, 2$. It is straightforward to calculate the probabilities of observing the four possible outcomes of the two endpoints from the given configuration. The result is shown in Table 6. Notice that V_j must satisfy the following inequalities to ensure non-negative probabilities:

$$\begin{aligned} & V_{jl} \leq V_j \leq V_{ju}\text{, where} \\ & V_{jl} = \max \{ - [P_{1j}P_{2j}(1-P_{1j})(1-P_{2j})]^{1/2}, - P_{1j}P_{2j}, \\ & \qquad - (1-P_{1j})(1-P_{2j}) \}\text{, and} \\ & V_{ju} = \min \{ [P_{1j}P_{2j}(1-P_{1j})(1-P_{2j})]^{1/2}, P_{1j}(1-P_{2j}), \\ & \qquad P_{2j}(1-P_{1j}) \} . \end{aligned} \tag{7}$$

We assume that the two treatment groups are independent and that the group sizes are known, then we have two independent multinomial distributions. A 4 x 2 table can be formed by taking an observation from each multinomial distribution. It is possible to observe 4 x 2 tables that have zero row margin in one or both of the corresponding 2 x 2 tables. If only one of the observed 2 x 2 tables contains a zero row margin, the test statistic is defined as the asymptotic p-value of the other 2 x 2 table. If both 2 x 2 tables contain zero row margin, the test can not be done because the individual p-values are not defined. We define such 4 x 2 tables as non-testable.

For the given response rates and covariances of the two responses, the power is defined as the probability of observing 4 x 2 tables, after excluding the non-testable tables, on which the overall null hypothesis can be rejected by our test procedure. The exact power can be calculated by exhausting all possible outcomes of the two multinomial distributions and by summing up the product of the two multinomial probabilities that the corresponding 4 x 2 tables can be rejected under a pre-determined level α, then divided by the probability of observing testable tables. That is,

$$\text{Power} = \frac{\sum\limits_{\text{adj-p} \leq \alpha} \Pr_1 \cdot \Pr_2}{1 - \Pr(\text{ non-testable 4 x 2 tables})}, \tag{8}$$

where $\Pr_j$ is the multinomial probability of observing the outcomes of group j, $j = 1, 2$.

The group sizes are sometimes so large that calculation of the exact power becomes infeasible. The power can be estimated by taking random samples from the two independent multinomial distributions and by calculating the proportion of rejected tables, after deleting non-testable 4 x 2 tables.

Table 6: Configurations and the corresponding multinomial probabilities

Tumor	Control	Treated
A	P_{11}	P_{12}
B	P_{21}	P_{22}
Cov	V_1	V_2

⇓

Tumor	Group 1	Group 2
No	$(1-P_{11})(1-P_{21}) + V_1$	$(1-P_{12})(1-P_{22}) + V_2$
A only	$P_{11}(1-P_{21}) - V_1$	$P_{12}(1-P_{22}) - V_2$
B only	$(1-P_{11})P_{21} - V_1$	$(1-P_{12})P_{22} - V_2$
A and B	$P_{11} \cdot P_{21} + V_1$	$P_{12} \cdot P_{22} + V_2$

Table 7: Configurations and the corresponding multinomial probabilities

Tumor	Control	Treated
A	0.04	0.16
B	0.02	0.10
Cov	0.0192	0.024
Correlation	0.70	0.22

⇓

Tumor	Group 1	Group 2
No	0.96	0.78
A only	0.02	0.12
B only	0.00	0.06
A and B	0.02	0.04

Example 1 (*continued*)

Suppose the tumor incidence rates at site A and B and their correlations are given in Table 7. Following Table 6, the marginal probabilities of the two independent multinomial distributions can easily be derived. Notice that the expectation for each combination, using group size of 50, is exactly the one observed in Table 1. The powers under level 0.05 using Rom (1992) and the proposed procedures are estimated by Monte Carlo simulation with 5,000 samples. The results are displayed in Table 8. Both procedures have similar power in this example. Running on VAX 6000-620, the CPU time used by the proposed procedure is only about 5 minutes, in contrast to 1 hour and 44 minutes used by Rom's procedure.

Table 8: Powers of Rom (1992) and the proposed procedures

Method	Power	Confidence Interval	# of samples
Rom	0.697	(0.685, 0.710)	5,000
Proposed	0.699	(0.687, 0.712)	5,000

CONCLUDING REMARK

Our proposed procedure can easily be extended to ordered multinomial response by evaluating the asymptotic p-values of the r_i x c tables instead of 2 x c tables, where r_i is the number of possible outcomes at the *i*-th endpoint.

Although the powers of the proposed procedure under different configurations are not reported here, from our experience, the procedure has the best power when the asymptotic p-values are positively correlated, or, if several (all) endpoints are affected by the treatment.

REFERENCES

1. Agresti, A. (1990), *Categorical Data Analysis*, Wiley, New York.
2. Boyett, J. M. (1979), "Algorithms AS 144. Random RXC tables with given row and columns totals," *Appl. Statist* **28**, 329-332.
3. Brown, C.C. and Fears, T.R. (1981), "Exact significance levels for multiple binomial testing with application to carcinogenicity screens," *Biometrics*, **37**, 763-774.
4. Gail M., Mantel N. (1977), "Counting the Number of R X C Contingency Tables with Fixed Margins," *Journal of American Statistical Association*, **72** (360), 859-862.
5. Heyse, J. F. and Rom, D. M. (1988), "Adjusting for multiplicity of statistical tests in the analysis of carcinogenicity studies," *Biometrical Journal*, **8**, 883-896.
6. Hochberg, Y. and Tamhane, A. C. (1987), *Multiple Comparison Procedures*, Wiley, New York.
7. Mantel, N. (1963), "Chi-square tests with one degree of freedom: extensions of the Mantel-Haenszel procedure," *Journal of American Statistical Association*, **58**, 690-700.
8. Marcus, R., Peritz, E., and Gabriel, K. R. (1976), "On closed testing procedures with special reference to ordered analysis of variance," *Biometrika*, **63**, 655-660.
9. Rom, D. M. (1992), "Strengthening some common multiple test procedures for discrete data," *Statistics in Medicine*, **2**, 511-514.
10. Tarone, R. E. (1990), "A modified Bonferroni method for discrete data', *Biometrics*," **46**, 515-522.
11. Tukey, J. W., Ciminera, J. L., and Heyse, J. F. (1985), " Testing the statistical certainty of a response to increasing doses of a drug," *Biometrics*, **41**, 295-301.
12. Verbeek, A. and Kroonenberg, P. (1985), "A survey of algorithms for exact distributions of test statistics in *r* x *c* tables with fixed margins," *Computational Statistics & Data Analysis*, **3**, 159-185.
13. Westfall, P.H. (1985) "Simultaneous Small-sample Multivariate Bernoulli Confidence Intervals," *Biometrics*, **41**, 1001-1013.
14. Westfall, P. H. and Young (1989), S. S. "P-value adjustments for multiple tests in multivariate binomial models," *Journal of the American Statistical Association*, **84**, 780-786.

On a Partial Cross Validation in Nonparametric Regression

Andrzej S. Kozek
Department of Statistics, Macquarie University,
Sydney NSW 2109, Australia

Abstract

The number of calculations in the classical Cross Validation (CV) method grows very fast with the size of the sample. Hence various methods reducing the number of necessary calculations have been proposed: a Monte Carlo Cross Validation approximation [4], [5]; WARP-ing [10],[17]; and Binning [9]. In the present paper we discuss a new approach, Partial Cross Validation (PCV), saving on the computational effort while choosing the optimal smoothing parameter. In classical CV and in Generalized Cross Validation (GCV) it is necessary to calculate the sum of n squares of differences between Y_i and the leave-one-out estimates of the regression function at X_i. In PCV this sum is calculated only over a relatively small number k_n of properly chosen indices i. By choosing PCV-optimal window width we end up with both window width and estimator very close to their GCV-competitors. In Section 4 we present performance of the PCV and GCV methods in simulations with $n = 100$ and $k_n = 8$. In Section 3 we find conditions under which PCV has the same feature as GCV: it is, up to a constant, an unbiased estimator of the Mean Integrated Square Error (MISE).

1 Introduction.

One of the simplest representations of the regression function is given by

$$Y = r(X) + e, \tag{1}$$

where X and e are independent, $E(e) = 0$, and $Var(e) = \sigma^2 < \infty$. In case our information about $r(\cdot)$ is poor, e.g. when we know no adequate parametric model to which $r(\cdot)$ belongs, nonparametric methods provide reliable estimation tools. Nonparametric estimators of $r(x)$ based on i.i.d. observations (X_i, Y_i), $i = 1, \cdots, n$ considered in the literature include Nadaraya-Watson, k-th Nearest Neighbor, p-th Optimal Quantile, spline estimators, Gasser-Müller, LoEss, Local Polynomial, Local Parametric, and we refer for more comprehensive references to [3], [8], [11], [12], and [15].

Users of nonparametric methods must pay for the universal consistency with a slower rate of convergence and, so far, with much greater computational complexity. Methods called Randomized Cross Validation [4],[5], WARP-ing [10], [17], and Binning [9] considerably reduce (from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$) the costs of calculation of the estimator which are related to the necessity of multiple evaluation of the kernel. In the present paper we consider an application of Numerical Analysis to the estimation of the optimal smoothing parameter of nonparametric regression estimators. We propose PCV, a modification of CV and GCV (see [1]), by appropriate skipping over most of the terms in the original formula and weighting the remaining ones. The main idea consits in approximating an integral (Integrated Square Error (ISE))by using some of the standard methods available in the Numerical Integration Theory and then approximating the knots of integration by the closest points from the sample. This approach seems applicable also in the multivariate case, in nonparametric density estimation, in spline estimation, and in tomography [4], [5],[14] as well. We shall not pursue generality here and concentrate on presenting the method in case of the Nadaraya-Watson estimator. It is clear that especially the Binning can also be incorporated into the methodology. However for the sake of simplicity of presentation we shell refer here direct to kernels. The idea of PCV has been to our knowledge first implemented in tomography [14] in the version of approximation of order one (see the rectangular version of the PCV listed at the end of section 2). The experience shows that it works reasonably well, at least for small sample sizes.

In Section 3 we show that for higher order ISE approximations $PCV_n(h)$ is asymptotically an unbiased estimator of $MISE_n(h)$, see Theorem 2. We implemented both versions of PCV in nonparametric and user friendly Fit Short 2.4 package, cf. [15]. A comparison of the performance of $GCV_n(h)$ and $PCV_n(h)$ in simulations is reported in Section 4.

2 The Partial Cross Validation

Consider the Nadaraya-Watson estimator of $r(X)$ which is given by

$$\hat{r}_h(x) = \frac{\sum_{i=1}^n Y_i\, K\left(\frac{x-X_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x-X_i}{h}\right)} = \sum_{i=1}^n \frac{1}{n} W_{n,h}\,(X_i)\cdot Y_i. \tag{2}$$

Estimator $\hat{r}_h(x)$ depends on a smoothing parameter h called a window width. The quality of the estimator strongly depends on the proper choice of the window width and is measured by the Mean Integrated Squared Error (MISE) given by

$$MISE_n(h) \;=\; E\,(ISE_n(h)), \tag{3}$$

where

$$ISE_n(h) \;=\; \int (\hat{r}_h(x) - r(x))^2 w(x) f(x) dx, \tag{4}$$

$f(\cdot)$ is the density function of X, and $w(.)$ is the indicator function of an interval $\mathcal{A}$, such that

$$\gamma < f(x) < \frac{1}{\gamma} \quad \text{for some } \gamma > 0 \text{ and for every } x \in \mathcal{A}.$$

A related random measure of discrepancy between $\hat{r}_h(x)$ and $r(x)$ is given by the Averaged Squared Error (ASE)

$$ASE_n(h) \;=\; \frac{1}{n}\sum (\hat{r}_h(X_j) - r(X_j))^2\, w(X_j).$$

ASE and ISE have been proved asymptotically equivalent to MISE [6], [7], [13],[10], [16], and the problem consists in finding $h = h_n(X_1,\cdots,X_n)$ minimizing any of them. Despite of a range of competitors (cf. [8]) CV-type methods are among the most popular in finding asymptotically optimal h. The original $CV_n(h)$ is given by

$$CV_n(h) \;=\; \frac{1}{n}\sum_{i=1}^n \left(Y_i - \hat{r}_h^{(i)}(X_i)\right)^2, \tag{5}$$

where $\hat{r}_h^{(i)}(x)$ is the leave-one-out estimator. CV admits some generalizations, here we shall refer to the GCV in the form discussed in [11],[12]:

$$GCV_n(h) \;=\; \frac{1}{n}\sum_{i=1}^n (Y_i - \hat{r}_h(X_i))^2 \cdot \Xi_i \cdot w(X_i), \tag{6}$$

where

$$\begin{aligned}\Xi_i &= \Xi\left(\frac{K(0)}{\sum_j K\left(\frac{X_i-X_j}{h}\right)}\right),\\ \Xi(u) &= 1+2u+\mathcal{O}(u^2),\;\; (u\to 0).\end{aligned}$$

PCV, a simple modification of the GCV has empirically determined computational complexity $\mathcal{O}(n)$ (as implemented in the package Fit Short 2.4). Let

$$PCV_n(h) \;=\; \sum_{i=1}^{k_n} \mu_i\cdot(Y_{i^\bullet} - \hat{r}_h(X_{i^\bullet}))^2 \cdot \Xi(X_{i^*})\cdot w(X_{i^*}), \tag{7}$$

where k_n is the number of components, μ_i are the weights, and i^* is a function of argument i from $\{1,\ldots,k_n\}$ into $\{1,\ldots,n\}$.

We shall need the following definitions and notation (cf. [2], pp.57 and 75).

Definition 1 *A numerical integration method of the form*

$$\int_a^b u(x)dx = \sum_{i=1}^m u(x_i)\cdot w_i + R_m(u) \tag{8}$$

is said to be of order s in a class of functions $\mathcal{F}$ $s+1$ times differentiable on $\mathcal{A}$ if $R_m(u) = \mathcal{O}(m^{-s})$ for every $u \in \mathcal{F}$.

Definition 2 *A numerical integration method is called a compound k_n-points Gauss rule with $k = m\cdot p$ if it results from from dividing the interval of integration into m equal subintervals and applying the p-point Gauss method to each of them.*

In applying any numerical integration rule to approximate an expected value of $g(X)$ we shall use representation

$$E_F\, g(X) \;=\; \int_0^1 g(F^{-1}(x))dx \tag{9}$$

and apply the numerical integration rule to the right hand side expression or, equivalently, transform the original knots x_i into the corresponding quantiles of the probability distribution function F. In what follows we shall assume that all necessary regularity and smoothness assumptions required in theorems in [2] on pp.57 and 75 are fulfilled.

Let $(X_1,Y_1),\ldots,(X_n,Y_n)$ be a given sample of independent pairs of random variables. We order them according to the increasing values of X's (with ties broken by the chronological order) getting $(X_{(1)},Y_{(1)},\ldots,X_{(n)},Y_{(n)})$ with $X_{(1)} \le X_{(2)} \le \ldots X_{(n)}$. Let (j) denote the index in the original sample corresponding to the $j-th$ order statistic and $[\alpha]$

be the best integer approximation of the real number α, rounding towards zero in case of ambiguity.

Below we list two versions we used in simulations. The first one corresponds to the 'rectangular' rule of numerical integration. We do not know if Theorem 2 holds true for this numerical integration rule. The second one corresponds to the compound k_n-points Gauss rule of order $2p$ with $k_n = m_n \cdot p$, $m_n = n^{\frac{1}{2p+1}}$, and we will show in Section 3 that it is, up to a constant, an unbiased estimator of MISE.

1. *a* rectangular *version of the PCV:* $k_n = c \cdot \log(n)$, $\mu_i = \frac{1}{k_n}$ for every i, and $i^* = \left(\left[\frac{i \cdot n}{k_n}\right]\right)$,
2. a compound gaussian *version of the PCV:* $k_n = n^{\frac{1}{2p+1}} \cdot p$, $p \geq 3$, μ_i are the weights in the k_n-points compound Gauss numerical integration method, $i^* = \left(\left[\frac{n+1}{2} + \frac{n-1}{2} \cdot x_i\right]\right)$, $i = 1, \ldots, k_n$, and x_i's are the i-th ordered abscissas of the k_n-knots of the compound Gauss integration rule on [-1,1].

3 Main results

The ISE given by (4) is an integral over interval $\mathcal{A}$, on which the density function $f(x)$ is positive. Hence for large n and smooth kernel K the estimator $\hat{r}_h(x)$ is well defined and smooth. So, if the regression function is also smooth the integrand of the ISE is a smooth function and the integral can be approximated with the use of numerical methods of integration. We shall pursue this program while paying attention to retain the proper order of approximation.

Let us approximate $ISE = ISE_n$ uniformly for $h \in H = h = [\frac{1}{C}n^{-\delta}, Cn^{-1+\delta}]$ for some $C > 0$ and $\delta > 0$ using a compound k_n,-point integration method of order s, $k_n = p \cdot m_n$, and then approximate the knots x_i by corresponding the closest sample points X_{i^*}. We have

$$\begin{aligned} ISE_n(h) &= \int g(x)w(x)f(x)dx \\ &= \sum_{j=1}^{m_n} \mu_j \cdot g(x_j) + \mathcal{O}_p(\frac{1}{m_n})^s \\ &= \sum_{j=1}^{m_n} \mu_j \cdot g(X_j^*) + \mathcal{O}_p(\frac{1}{m_n})^s + \mathcal{O}_p(\frac{m_n}{n}) \\ &= PISE_n(h) + \mathcal{O}_p(\frac{1}{m_n})^s + \mathcal{O}_p(\frac{m_n}{n}) \end{aligned}$$

The first equality follows just from the property of the numerical integration method, see [2] pp. 58 and 75, while the second one is implied e.g. by the Bahadur representation of quantiles [18]. To minimize the order of the error of approximation we choose $m_n = n^{\frac{1}{s+1}}$ yielding the error of order $\mathcal{O}(n^{-\frac{s}{s+1}})$. To keep the error on level $o(n^{-\frac{4}{5}})$ we take $s > 4$. So, we get

Theorem 1 *If the estimator $\hat{r}_n(x)$ and the regression function $r(x)$ belong to $\mathcal{F}$ and the integration method is of order s on $\mathcal{F}$ then for $s > 4$ we have for $h \in H$*

$$ISE_n(h) = PISE_n(h) + o_p(n^{-\frac{4}{5}}). \tag{10}$$

In the next step we consider relations between PISE and PCV assuming the same m_n in both cases and also the same index selection method i^*. Indeed we have

$$\begin{aligned} PISE_n(h) &= \sum_{j=1}^{m_n} \mu_{j^*} \left((\hat{r}_h - r)^2\right) \left(X_{j^*}\right) \cdot \Xi_{j^*} \cdot w(X_{j^*}) \\ &= \sum_{j=1}^{m_n} \mu_j \cdot \left(Y_{j^*} - \hat{r}_h(X_{j^*})\right)^2 \cdot \Xi(X_{j^*}) \cdot w(X_{j^*}) \\ &+ \sum_{j=1}^{m_n} \mu_j \cdot e^2_{j^*} \cdot \Xi(X_{j^*}) \cdot w(X_{j^*}) \\ &+ 2 \sum_{j=1}^{m_n} \mu_j \cdot e_{j^*} \cdot \left(Y_{j^*} - \hat{r}_h(X_{j^*})\right) \cdot \Xi(X_{j^*}) \cdot w(X_{j^*}) \\ &= PCV_n(h) + \sum_{j=1}^{m_n} \mu_j \cdot e^2_{j^*} \cdot w(X_{j^*}) \\ &+ 2 \sum_{j=1}^{m_n} \mu_j \cdot e_{j^*} \cdot \left(Y_{j^*} - \hat{r}_h(X_{j^*})\right) \cdot w(X_{j^*}) \\ &+ \frac{2}{n} \sum_{j=1}^{m_n} \mu_j \cdot e^2_{j^*} \cdot w(X_{j^*}) + \mathcal{O}_p\left(n^{-\frac{4}{5}}\right) \\ &= PCV_n(h) + T_1 + T_2 + T_3 + \mathcal{O}_p\left(n^{-\frac{4}{5}}\right). \end{aligned}$$

T_1 does not depend on h while in a way similar to [12], p. 154-155 one can verify that

$$E\left(T_2 | X_1, \ldots, X_n\right) = -T_3. \tag{11}$$

Hence we get the following theorem.

Theorem 2 *Under the assumptions of Theorem 1 $PCV_n(h)$ given by (7) is, up to a constant, an unbiased estimator of $MISE_n(h)$, i.e. for $h \in H$,*

$$\begin{aligned} E\,PCV_n(h) &= MISE_n(h) \\ &+ E\left(\sum_{j=1}^{m_n} \mu_j \cdot e^2_{j^*} \cdot w\left(X_{j^*}\right)\right) \\ &+ o(n^{-4/5}) \end{aligned} \tag{12}$$

Theorem 2 suggests that arguments $\bar{h}_n$ minimizing PCV_n can be used in 2 instead of $\hat{h}_n$ minimizing GCV_n. It is plausible that paralleling arguments in [6] or [13] one can show optimality of the $\bar{h}_n$ minimizing $PCV_n(h)$ in the sense of minimizing $MISE_n(h)$. However detailed verification of this conjecture is beyond the scope of the present note.

4 Simulations

Using package Fit Short 2.42 we compared PCV and GCV on many both simulated and real data. In general, the behavior of the PCV is on the level of CV and GCV with very often only minor differences in estimators from these methods. We shortly report here on two typical simulations. In both cases we applied 8-point Gauss integration method with $n = 100$, $m_n = 1$, and $k_n = p = 8$.

1. $r(x) = (sin(2\pi x^3))^3$, X's uniform on $[0, 1]$, $e \sim N(0, \sigma = 0.7)$, see Figures 1 and 2,

2. $r(x) = T_4(x) = 8x^4 - 8x^2 + 1$, X's uniform on $[0, 1]$, $e \sim N(0, \sigma = 0.7)$, see Figures 3 and 4.

In the former case we have almost identical resulting estimators of the regression curves, in the latter one $\bar{h}_{PCV}$ oversmoothes the regression curve. The regression function in 1 was considered by Härdle in [12] for $n = 256$ and $\sigma^2 = 0.5$.

References

[1] Craven, P. and Wahba, G. (1979). "Smoothing noisy data with spline functions". *Numer. Math.* **31**, 377-403.

[2] Davis, P.J. and Rabinowitz P. *Methods of Numerical Integration*, 1975. Academic Press.

[3] Fan, J. (1992). "Local linear regression smoothers and their minimax efficiency". *The Annals of Statistics* **21**, 196-216.

[4] Girard, D. (1992). Comment on "Empirical functionals and efficient smoothing parameter selection" by Hall, P. and Johnstone, I. *J. R. Statist. Soc.* B **54** 521.

[5] Girard, D. (1994). "The fast Monte-Carlo cross-validation and C_L procedures: Comments, new results and application to image recovery problems". To appear in *Comp. Stat.*

[6] Hall, P. (1984a). "Asymptotic properties of integrated square error and cross-validation for kernel estimation of a regression function". *Z. Wahrsch. Verw. Gebiete* **67**, 175-196.

[7] Hall, P. (1984b). "Integrated square error properties of kernel estimators of regression functions". *The Annals of Statistics*, **12**, 241-260.

[8] Hall, P. and Johnstone I. (1992). "Empirical Functionals and Efficient Smoothing Parameter Selection ". *J. R. Statist. Soc.* B, **54**, 475-530.

[9] Hall, P. and Wand, M.P. (1993). "On the Accuracy of Binned Kernel Density Estimators". Working Paper Series 93-003, The Univ. of NSW, Austr. Grad. School of Management.

[10] Härdle, W. (1986). "Approximations to the Mean Integrated Squared Error with Applications to Optimal Bandwidth Selection for Nonparametric Regression Function Estimators". *J. of Multivariate Analysis* **18**, 150-168.

[11] Härdle, W. *Applied Nonparametric Regression*, 1990. Cambridge University Press.

[12] Härdle, W. *Smoothing Techniques with implementation in S*, 1991. Springer-Verlag.

[13] Härdle, W. and Marron, J.S. (1985). "Optimal bandwidth selection in nonparametric regression function estimation", *The Annals of Statistics*, **13**, 1465-81.

[14] Hudson, M. and Lee, C.M. (1994). "Deblurring Images Subject to Poisson Variability" - a paper presented at *Statistics'93 Sept. 27-Oct. 1, 1993*, Wollongong.

[15] Kozek, A.S. (1992). "A New Nonparametric Estimation Method: Local and Nonlinear".*Computer Science and Statistics: Proc. of the 24th Symp. on the Interface*, 389-39.

[16] Marron, J. S., and Härdle, W. (1986). "Random approximations to an error criterion of nonparametric statistics". *Journal of Multivariate Analysis*, **20** 91-113.

[17] Scott, D.W. (1985). "Averaged shifted histograms: effective nonparametric density estimators in several dimensions". *The Annals of Statistics* **13**, 1024-1040.

[18] Serfling, R.J. *Approximation Theorems of Mathematical Statistics*, 1980. Wiley & Sons.

[19] Stone, C.J. (1984). "Cross-validatory choice and assessment of statistical predictions ". *J. R.Statist. Soc.* B, **36**, 111-147.

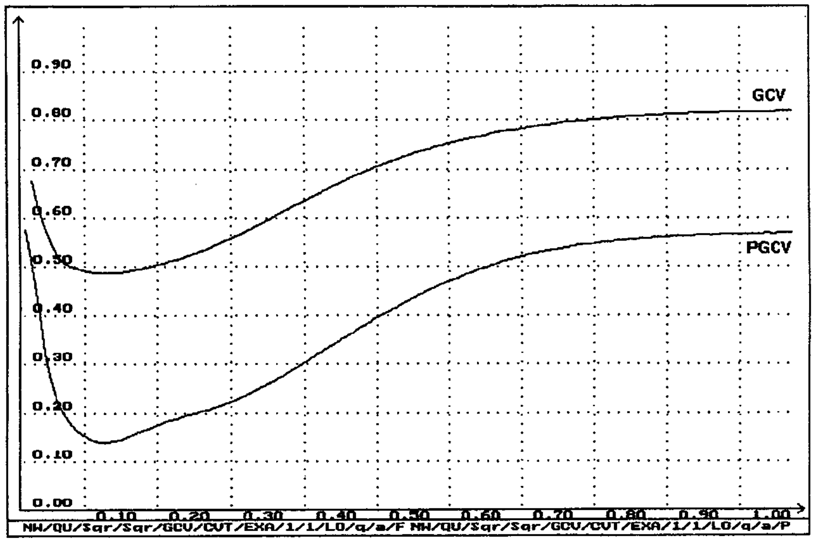


Fig 1. Graphs of GCV(h) and PGCV(h), n=100, $r(x)=\sin(2\pi x^3)^3$.

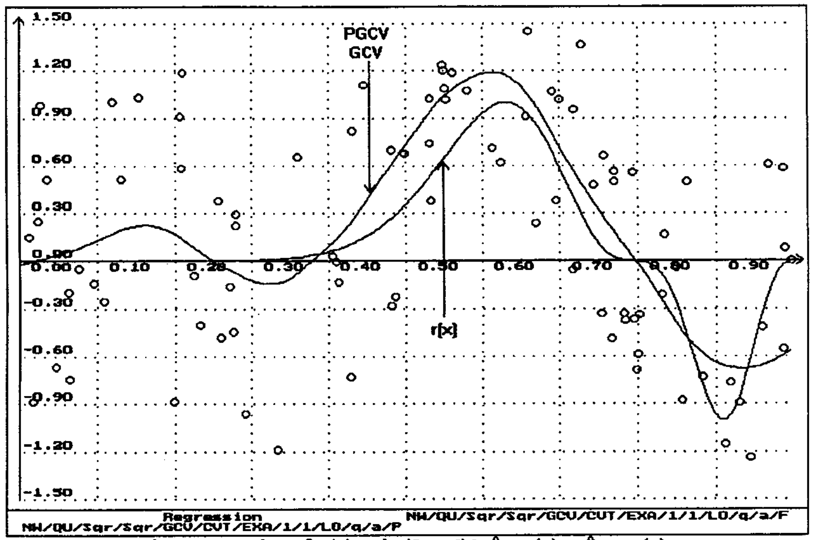


Fig 2. Graphs of $r(x)=\sin(2\pi x^3)^3$, $\hat{r}_{GCV}(x) \approx \hat{r}_{PGCV}(x)$.

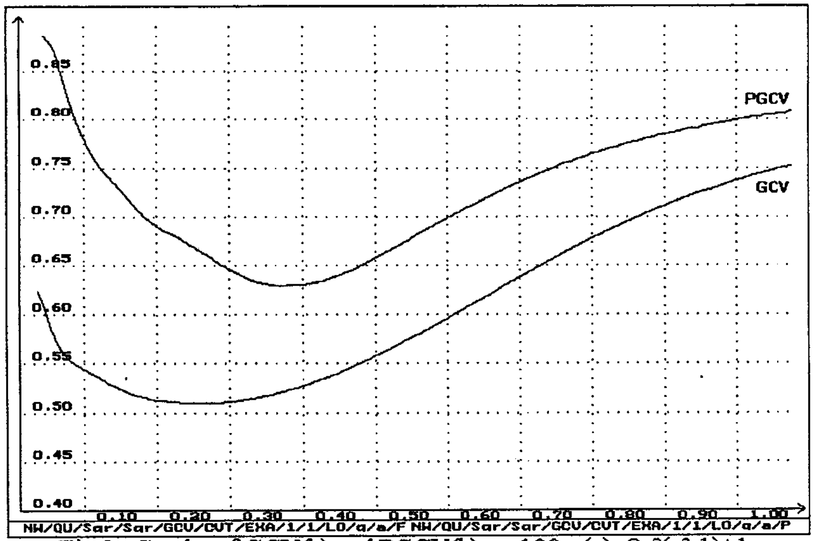


Fig 3. Graphs of GCV(h) and PGCV(h), n=100, $r(x)=8x^2(x^2-1)+1$.

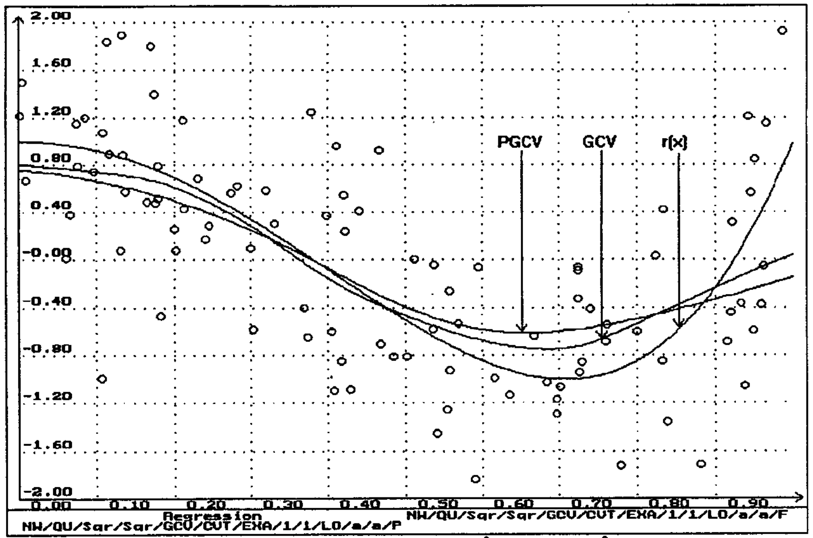


Fig 4. Graph of $r(x)=8x^2(x^2-1)+1$, $\hat{r}_{GCV}(x)$ and $\hat{r}_{PGCV}(x)$.

AN ITERATIVE PROJECTION METHOD FOR NONPARAMETRIC ADDITIVE REGRESSION MODELLING

M. G. Schimek[1], H. Stettner[2] and J. Haberl[2]

[1] Medical Biometrics Group, University of Graz Medical Schools, A-8036 Graz, Austria, Europe
[2] Department of Mathematics and Statistics, University of Klagenfurt, A-9020 Klagenfurt, Austria, Europe

Abstract

For the estimation of additive regression models in a nonparametric fashion based on some linear scatterplot smoother the solution of large linear, often ill-posed, systems is required. Standard iterative approaches of the Jacobi and Gauss-Seidel type only apply to non-singular system matrices, although their use for ill-posed problems is most common. In this paper an iterative projection method with some favourable properties is proposed: Convergence can be established without restrictions on the system matrix. For singular systems an optimal solution can be obtained. Finally, it is possible to take advantage of the shape of specific system matrices when calculating the solution.

1. Introduction and motivation

Projection pursuit regression (FRIEDMAN and STUETZLE, 1981) and generalized additive models (HASTIE and TIBSHIRANI, 1990) are well-known examples of non-parametric regression problems with scatterplot smoothers. These approaches require solving large linear equation systems. To reduce the computational costs the so-called backfitting algorithm was introduced, a numerical procedure related to Jacobi and Gauss-Seidel iteration. The basic idea is to determine estimates for the covariates successively in a non-parametric manner (scatterplot smoother). Backfitting uses currently available information from all covariates, except the covariate of which the estimates are just computed. This leads to a splitting of the system matrix into *d* blocks, each block corresponding to one of the predictor variables X_j, $j=1, 2, \dots ,d$. Finally an iterative procedure, most often Gauss-Seidel is applied to these blocks. Relaxation can improve the speed of convergence, but is usually not implemented in statistical software. For a discussion of iterative procedures to solve linear equation systems in the context of additive regression modelling see SCHIMEK, NEUBAUER and STETTNER (1994).

Although there are reports that backfitting works well (e.g. BUJA, HASTIE, and TIBSHIRANI, 1989) in most situations alternative procedures should be considered. First of all, Jacobi and Gauss-Seidel iteration as well as variants of it were not developed for solving (nearly) singular systems. In non-parametric regression linear scatterplot smoothers such as spline and kernel techniques are most common. The smoothed data are design-dependent (number and location of knots, smoothing parameter, kernel characteristics and bandwidth), hence ill-posed or singularity problems must be expected and in principle the associated normal equation system should not be solved by standard algorithms. Further we have to be aware of concurvity, also contributing to the singularity of the system matrix. As a direct consequence we cannot predict the speed of convergence and the quality of the obtained results.

Direct, non-iterative procedures could be applied to such singular normal equation systems. SCHIMEK, STETTNER, and HABERL (1992) proposed a Tichonow regularization technique. It yields exact solutions on the one hand. On the other hand it is too expensive for routine use. Tichonow regularization is rather a valuable tool for the comparison of results obtained by other numerical concepts.

In this paper we propose an alternative procedure with a number of favourable properties. The idea is to obtain correct solutions for large linear systems in an iterative, cheap manner, even when the system matrix is singular. For that purpose we take a projection-oriented, geometrically motivated approach.

2. An iterative projection method

The iterative projection method we want to develop is related to a row-oriented procedure introduced by KACZMARZ (1937) and a column-oriented technique due to de la GARZA (1951, see also HACKBUSCH, 1991, p. 203 and HOUSEHOLDER, 1975, section 4.2). We assume a sequence of iterative projections which forms an "instationary process" in the terminology of MAESS (1988, p. 116).

Let us have a linear equation system $\mathbf{Ax} = \mathbf{b}$ to be solved in $\mathbf{x}$ with $\mathbf{A}$ a $n \times n$ matrix, $\mathbf{x}$ and $\mathbf{b}$ n-dimensional vectors. Further we define $\mathbf{A} = (\mathbf{a}(1), \mathbf{a}(2), ..., \mathbf{a}(n))$ where $\mathbf{a}(i)$ denotes the i-th column vector of $\mathbf{A}$. We represent $\mathbf{b}$ step by step via a sequence of the form

$$\mathbf{b} = \sum_{i=1}^{k} \mu(i)\mathbf{a}'(i) + \mathbf{u}(k) \qquad (1)$$

where i=1, 2, ... , k, k=1, 2, ... , and

$$\mathbf{a}'(i) = \mathbf{a}^*((i-1) \bmod n + 1).$$

The $\mathbf{a}^*(1), \mathbf{a}^*(2), ..., \mathbf{a}^*(n)$ are a permutation of the $\mathbf{a}(1), \mathbf{a}(2), ..., \mathbf{a}(n)$ to improve the convergence speed (compare with the "cyclical criterion" described in MURTY, 1983, p.457). The vector $\mathbf{u}(k)$ represents the "unexplained" component of $\mathbf{b}$ and is the perpendicular from $\mathbf{u}$(k-1) to the dimension $\mathbf{a}'(k)$ at iteration step k. The coefficients $\mu(i)$ are determined in each step by an optimality criterion

$$f(k, \mu(k); \mathbf{a}'(1), \mathbf{a}'(2), ..., \mathbf{a}'(k)) = 0$$

they have to fulfil (e.g. require that $\mathbf{u}(k)$ is the perpendicular of $\mathbf{u}(k\text{-}1)$ onto $\mathbf{a}'(k)$. We can establish conditions for the optimality criterion under which $\mathbf{u}(k)$ converges towards 0.

For the evaluation of the coefficients $\mu(i)$ we can take advantage of structural features of the system matrix $\mathbf{A}$. This is an important aspect when solving the normal equations associated with additive regression models. According to BUJA, HASTIE, and TIBSHIRANI (1989, p.477) we have to solve the system

$$\begin{pmatrix} \mathbf{I} & \mathbf{S}_1 & \mathbf{S}_1 & \cdots & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{I} & \mathbf{S}_2 & \cdots & \mathbf{S}_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{S}_d & \mathbf{S}_d & \mathbf{S}_d & \cdots & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_d \end{pmatrix} = \begin{pmatrix} \mathbf{S}_1\mathbf{y} \\ \mathbf{S}_2\mathbf{y} \\ \vdots \\ \mathbf{S}_d\mathbf{y} \end{pmatrix}$$

in our notation $\mathbf{Ax} =$ $\mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are block matrices of smoothing operators (matrices) $\mathbf{S}_l$, $\mathbf{x}_l$ solution vectors and $\mathbf{y}$ a dependent variable vector in an additive regression model.

3. Features of the iterative projection method

There are a number of advantages of the proposed method:

- It always converges because convergence does not depend on the characteristics of the system matrix **A**, such as diagonal dominance.
- For singular systems an optimal solution can be obtained.
- The shape of specific system matrices **A** (e.g. due to certain scatterplot smoothers like cubic smoothing splines) can be exploited for the calculation of the solution.

As disadvantage has to be mentionend:

- Slow convergence in its standard version (see e.g. HACKBUSCH, 1991, p.204 for the Kaczmarz procedure).

To overcome this weak point of the iterative projection method two approaches can be taken: The one is to introduce a projection-specific relaxation concept. The other is to resort to parallel processing.

4. Proof of convergence

We prove convergence for the optimality criterion

$$\mu(k) = (\mathbf{u}(k\text{-}1), \mathbf{a}'(k)) / (\mathbf{a}'(k), \mathbf{a}'(k)),$$
$$\mathbf{a}^*(k) = \mathbf{a}(\mathrm{k}). \qquad (2)$$

In this situation equation (1) can be written as

$$\mathbf{b} = \sum_{j=1}^{n} \left\{ \sum_{\mathbf{a}'(i)=\mathbf{a}(j),\, i\le k} \mu(i) \right\} \mathbf{a}(j) + \mathbf{u}(k).$$

Aggregating in *m(j)* all *μ(i)* belonging to some *j* yields

$$\mathbf{b} = \sum_{j=1}^{n} m(j)\mathbf{a}(j) + \mathbf{u}(k)$$

and (2) takes the form

$$m(j) = \frac{(\mathbf{b},\mathbf{a}(j))}{(\mathbf{a}(j),\mathbf{a}(j))} - \sum_{\substack{l=1\\ l\ne j}}^{n} m(l)\frac{(\mathbf{a}(j),\mathbf{a}(l))}{(\mathbf{a}(j),\mathbf{a}(j))}. \qquad (3)$$

Formula (3) can be understood as an iterative solution of an equation system with the system matrix

$$\mathbf{H} = (h_{ij}) = ((\mathbf{a}(i),\mathbf{a}(j))).$$

For the convergence of this sequence we apply a classical theorem on the convergence of iterative procedures (see TODD, 1962, p. 222ff for details): Is some matrix **H** Hermitian and positive definite then the iterative procedure

$$\mathbf{x}(r{+}1) = \mathbf{d} + \mathbf{C}\,\mathbf{x}(r), \quad r = 0,1,2,\ldots$$

converges for arbitrary $\mathbf{x}(0)$ towards the solution of $\mathbf{Hx}=\mathbf{b}$, where

$$\mathbf{C} = -\mathbf{L}^{-1}\,\mathbf{U}$$

$$\mathbf{L} = (h_{ik}, i \ge k)$$
$$\mathbf{U} = (h_{ik}, i < k)$$
$$\mathbf{H} = \mathbf{L} + \mathbf{U}$$
$$\mathbf{d} = \mathbf{L}^{-1}\,\mathbf{b}.$$

As a direct result the approach in (1) is self-correcting and numerically stable. Numerical errors in step *k* are compensated during the computation of *m* in step *k+1.* These advantages are not shared with other procedures recalculating **x**(*k*) in each step.

5. Singular equation systems

Another important advantage of the proposed method is its convergence for singular equation systems. Let us have **b** not a member of the linar space spanned by the columns of **A**, and

$$\mathbf{b} = \mathbf{b}_A + \mathbf{b}_o$$

the unique partition of **b** with $\mathbf{b}_o$ orthogonal to the column space of **A**. For reason already given **u**(k) converges in

$$\mathbf{b}_A = \left\{ \sum_{a'(i)=a(j),\, j\le k} \mu(i) \right\} \mathbf{a}(j) + \mathbf{u}(k)$$

to 0. When calculating the μ*(i)* successively from **b** instead of $\mathbf{b}_A$, the same coefficients are obtained, because $\mathbf{b}_o$ does not contribute to the solution μ*(i)*:

$$\mu(v+1) = \frac{(\mathbf{b}_o + \mathbf{u}(v),\mathbf{a}(v+1))}{(\mathbf{a}(v+1),\mathbf{a}(v+1))} = \frac{(\mathbf{u}(v),\mathbf{a}(v+1))}{(\mathbf{a}(v+1),\mathbf{a}(v+1))}.$$

Finally we have

$$\mathbf{b} - \sum \mu(i)\mathbf{a}(i) =: \mathbf{u}'(k) \to \mathbf{b}_o$$

with μ(*i*) forming the solution **x** of $\min \|\mathbf{Ax} - \mathbf{b}\|$.

6. The algorithm

The algorithm is simple in its structure and can be expressed as follows.

```
read $n$, a(), b, *
$k = 0$, x() = 0
repeat
  $k = k+1$
  a'(k) = a*(($k$-1)mod $n$ + 1)
  solve $f(k,\mu(k),\mathbf{a}'(1),\ldots,\mathbf{a}'(k)) = 0$
  update x()
until
```

```
    terminating condition = true
end
```

An interesting aspect of the algorithm is the possibility to develop it into a parallel processing procedure. The necessary computer architecture is characterised by a multiple instruction stream, single data stream organization (see e.g. KRISHNAMURTI and NARAHARI, 1993, p. 69f).

7. An illustrative example for standard and relaxed iterative projection solutions

Let us solve the equation $\mathbf{Ax} = \mathbf{b}$, where **A** does not have diagonal dominance. We assume **A** = [col(2,2,1), col(1,3,1), col(1,2,2)] and **b** = col(4,7,4). Applying the optimality criterion

$$\mu(k) = (\alpha \mathbf{u}(k-1), \mathbf{a}'(k)) \,/\, (\mathbf{a}'(k), \mathbf{a}'(k))$$

we obtain the standard solution for α = 1 and a relaxed solution for α = 1.2 (larger than one to improve the speed of convergence). *Table 1* displays the approximations $\hat{x}$ in comparison with the exact result **x** = col(1,1,1) for n = 50 and n = 100.

100 take about twice the time of 50 unrelaxed iterations. For the relaxed solution the computational costs are only a factor 1.1 higher (reference 50 iterations) but the precision of the obtained result is improved by a factor 2 - 4 . Hence the relaxation technique is quite promising and should be studied in more detail (i.e. in a simulation experiment).

Table 1: Results for standard and relaxed iterative projections

exact x	**approximations $\hat{\mathbf{x}}$**		
	$\alpha = 1, n = 50$	$\alpha = 1, n = 100$	$\alpha = 1.2, n = 50$
1	0.99981	0.99999	1.00005
1	0.99959	1.00000	1.00022
1	1.00058	0.99999	0.99974

8. References

BUJA, A., HASTIE, T. and TIBSHIRANI, R. (1989). Linear smoothers and additive models. *Ann. Statist. 17*, 453-510.

FRIEDMAN, J. H. and STUETZLE, W. (1981). Projection pursuit regression. *J. Am. Statist. Assoc. 76*, 817-823.

De la GARZA, A. (1951). An iterative method for solving linear equations. *Oak Ridge Gaseous Diffusion Plant, Rep. K-731*, Oak Ridge, TN.

HACKBUSCH, W. (1991). Iterative Lösung großer schwachbesetzter Gleichungssysteme. *Teubner*, Stuttgart.

HASTIE, T. and TIBSHIRANI, R. (1990). Generalized additive models. *Chapman and Hall*, London.

HOUSEHOLDER, A. S. (1975). The theory of matrices in numerical analysis. *Dover*, New York.

KACZMARZ, S. (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Internat. Acad. Polon. Sci, Cl.A 1937*, 355-357.

KRISHNAMURTI, R. and NARAHARI, B. (1993) Parallel computer architectures. In Rao, C. R. (ed.) Handbook of statistics 9 Computational statistics. *North Holland*, Amsterdam, 69-100.

MAESS, G. (1988). Projection methods solving rectangular systems of linear equations. *J. Comp. Appl. Math. 24*, 107-119.

MURTY, K. G. (1983). Linear programming. *Wiley*, New York.

SCHIMEK, M. G., NEUBAUER, G. P. and STETTNER, H. (1994). Backfitting and related procedures for non-parametric smoothing regression: A comparative view. In DUTTER, R. (ed.) COMPSTAT '94. Proceedings in Computational Statistics. *Physica*, Heidelberg. In press.

SCHIMEK, M. G., STETTNER, H. and HABERL, J. (1992). An operator method for backfitting with smoothing splines in additive models. In DODGE, Y. and WHITTAKER, J. (eds.) Computational Statistics. Volume I. *Physica*, Heidelberg, 487-491.

TANABE, K. (1971). Projection method for solving a singular system of linear equations and its applications. *Numer. Math. 17*, 203-214.

TODD, J. (ed.; 1962). Survey of numerical analysis. *McGraw Hill*, New York.

Reseach supported by "Forschungsfonds der Universität Klagenfurt".

Nonparametric Curve Estimation from Indirect Observations

By Sam Efromovich *
Department of Mathematics and Statistics, University of New Mexico
Albuquerque, New Mexico 87131

Abstract

A problem of nonparametric curve estimation from indirect observations is considered. Asymptotically optimal orthogonal series estimator is suggested for regular setting. For irregular setting a consistent estimator, which is also rate optimal for some familiar cases, is suggested as well. Particular applications are density, filtering and nonparametric regression deconvolution and nonparametric regression with errors in predictors.

1 Introduction

Consider a problem of estimating function $f : R \to R$ when only convolution $g(t) = \int f(x)k(t-x)dx$ of f with the known function k is available for direct statistical observation.

The familiar examples are: (i) Density deconvolution when one estimates density f of a random variable U based on n i.i.d. observations $X_1, \dots, X_n$ having the same distribution as that of X and $X = U + \epsilon$ where U and ϵ are independent and probability density k of the measurement error ϵ is given; (ii) Nonparametric blurred image reconstruction when one estimates f using n i.i.d. observations $\{(Y_l = g(t_l) + \xi_l, t_l),\ l = 1, \dots, n\}$, here ξ is the error; Nonparametric regression with errors in predictors when one observes n i.i.d. realizations of (Y, X) where $Y = f(U) + \xi$ and $X = U + \epsilon$.

For a regular case when the Fourier transformation $h_k(v) = \int \exp(ivt)k(t)dt$ does not vanish, i.e., $h_k(v) \neq 0$, the most relevant results to our research are obtained by Donoho and Low (1992), Fan (1991,1993) and Fan and Truong (1993) where rate optimal deconvolution kernel estimates are suggested for a wide varieties of settings. Particularly, for the density deconvolution Fan (1991,1993) suggests the following kernel estimator. Let $K(x)$ be a traditional kernel function and $h_K(v) = \int \exp(iux)K(x)dx$ be its Fourier transform with $h_K(0) = 1$. Then the deconvolution density estimator is

$$\tilde{f}_n(x) = (2\pi)^{-1} \int \exp(-ivx) h_K(vt_n) \hat{h}(v) h_g^{-1}(v) dv$$

for suitable choice of a bandwidth t_n, where

$$\hat{h}_X(v) = n^{-1} \sum_{l=1}^{n} \exp(ivX_l) \tag{1}$$

is the empirical characteristic function of X.

Fan (1991,1993) shows that this estimate is rate optimal (as sample size increases) for two important classes of distributions of ϵ. Namely, for supersmooth distributions of order β when the corresponding characteristic functions $h_\epsilon(v)$ of noise ϵ satisfy

$$d_0|v|^{\beta_0} \exp(-|v|^\beta/\gamma) \leq |h_\epsilon(v)| \leq d_1|v|^{\beta_1} \exp(-|v|^\beta/\gamma)$$

and for the ordinary smooth distributions of order β when the characteristic functions are not decaying and satisfy

$$d_0|v|^{-\beta} \leq |h_\epsilon(v)| \leq d_1|v|^{-\beta}$$

as $v \to \infty$, here d_0, d_1, β and γ are some positive constants and β_0 and β_1 are constants. The examples of supersmooth distributions are normal, mixture normal and Cauchy, the examples of ordinary smooth distributions are gamma and double exponential distribution.

Similar results, which again hold for these two classes of distributions, are known for the other settings, including nonparametric regression with errors in predictors (see Fan and Truong (1994)).

There are two main questions which will be addressed in this paper:

- What is the optimal risk convergence for arbitrary function k which Fourier transform does not vanish, in particular, for arbitrary $h_\epsilon(v) \neq 0$?

- Can we suggest an optimal estimate for irregular case when the Fourier transform of k vanishes?

To explore the problem we shall use the orthogonal series approach. Below a short heuristic explanation of

*This research supported by NSF Grant DMS-9123956

this approach is given for the model of density deconvolution.

Suppose that estimated density $f(u)$ is supported over a given finite interval, for instance $[0, 2\pi]$. Then, under some very mild assumptions on the estimated density f it may be approximated for different loss functions with a desired accuracy via appropriate choice of the array $\{\lambda_j; J\}$ by orthogonal series

$$f(u, J, \{\lambda_j\}) = (2\pi)^{-1} \sum_{|j| \leq J} \lambda_j h_U(j) \exp(-iju) \qquad (2)$$

where $h_U(v)$ is the characteristic function of the random variable U. Here $0 \leq \lambda_j \leq 1$ are the smoothing coefficients and J is a cutoff.

It is well known that for the independent U and ϵ the characteristic function of the sum $X = U + \epsilon$ is equal to the product of the characteristic functions of U and ϵ, that is, $h_X(v) = h_U(v)h_\epsilon(v)$.

Thus, using the empirical characteristic function $\hat{h}_X(v)$ defined in (1) as an estimate for $h_X(v)$ and assuming that h_ϵ does not vanish (recall that distribution of noise ϵ is known and therefore the characteristic function is known as well) we obtain an estimate

$$\hat{f}_n(u, J, \{\lambda_j\}) = (2\pi)^{-1} \sum_{|j| \leq J} \lambda_j \hat{h}_X(j) h_\epsilon^{-1}(j) \exp(-iju) \ . \qquad (3)$$

Hereafter $h^{-1}(v) = \bar{h}(v)/|h(v)|^2$ where $\bar{h}$ is the complex conjugate of h.

Surprisengly enough, we shall see that this estimate may be used for the other discussed statistical models as well with the only difference that instead of the empirical characteristic function we use the corresponding familiar estimates for the case of direct observations.

Now we are in a position to explain how to solve the deconvolution problem when $h_\epsilon(j)$ is equal to zero for some j; the familiar examples are uniform, triangle and lattice-valued ϵ. We restrict our attention to the case when there exist decaying as $m \to \infty$ sequences s_{jm} such that

$$h_\epsilon(j + s_{jm}) > 0 \qquad (4)$$

for every integer j. Notice that for deconvolution of an arbitrary density f such assumption is necessary for consistent estimation. Thus, one can estimate $h_f(j + s_{jm})$ rather than $h_f(j)$ and then use the continuity of $h_f(j)$. In this paper we will use this idea; slightly different approach, based on the L'Hopital's rule, is explored in Efromovich (1994).

Section 2 is devoted to rate- and sharp-optimal estimation for the regular setting. Using the modern approach, which maps the different models into filtering in white noise (see Brown and Low (1990)), we explore the problem on example of a signal recovery in white noise. In Section 3 the irregular setting is considered on example of density deconvolution. Some possible extensions are discussed in Section 4.

2 Optimal Signal Recovery

The considered problem is to recover a periodic signal $f(t)$ from an observation $Y_n(t)$ such that

$$dY_n(t) = (Kf)(dt) + n^{-1/2} dw(t) \ , \quad 0 \leq t \leq 2\pi \qquad (5)$$

where $w(t)$ is the Brownian motion ($dw(t)$ is a so-called white noise), $Kf = \check{f}$ is the given operator such that the Fourier transform of $\check{f}$ satisfies $\int_0^{2\pi} \exp(ivt)\check{f}(t)dt = h_f(v)h_K(v)$ where $h_f(v) = \int_0^{2\pi} f(t)\exp(ivt)dt$ and the function $h_K(v)$ is given. Our problem is to estimate the l-th derivative of f.

Let $\|f\|_p = [\int_0^{2\pi} |f(t)|^p dt]^{1/p}$ be the familiar L_p-norm of f where $1 \leq p < \infty$ and $\|f\|_\infty = ess \ \sup_{t \in [0,2\pi]}(f(t))$; let $f^{(l)}$ mean the l-th derivative and $\lfloor \alpha \rfloor$ be the integer part of the positive α.

Throughout the paper we always assume that f belongs to either Lipschitz class $Lip(\alpha)$ of periodic functions when the functions are $\lfloor \alpha \rfloor$-fold continuously differentiable and periodic on the circle $[0, 2\pi]$, $\|f\|_2 \leq A < \infty$ and $|f^{(\lfloor \alpha \rfloor)}(u) - f^{(\lfloor \alpha \rfloor)}(v)| < Q|u - v|^{\alpha - \lfloor \alpha \rfloor}$ for $u, v \in [0, 2\pi]$, or to a Sobolev $H(\alpha, Q)$ class of periodic square integrable functions such that the corresponding Fourier transformations $h_f(v) = \int_0^{2\pi} f(t)\exp(ivt)dt$ of f satisfy inequality $\sum_{j=-\infty}^{\infty}[1 + |j|^{2\alpha}]|h_f(j)|^2 \leq Q$.

We shall consider the Lipschitz classes of functions when rate optimal estimation is investigated and refer to the Sobolev classes when sharp optimal Mean Integrated Squared Error (MISE) convergence is explored.

Our assumption on periodicity of estimated function is not crucial for our approach but it is very convenient, the interested reader is also referred to discussion of aperiodicity in Efromovich (1994).

It is well known that (5) may be rewritten as an infinite array of discrete observations

$$Y_j = h_f(j)h_K(j) + n^{-1/2}\xi_j \ , \quad j = \ldots, -1, 0, 1, \ldots \qquad (6)$$

where $Y_j = \int_0^{2\pi} \exp(ijt)dY(t)$ and ξ_j are i.i.d. standard normal random variables.

Set

$$\delta_n^2(\alpha, Q, l) = \sum_{|j| < J_n} |j|^{2l} |h_K(j)|^{-2}(1 - (|j|/J_n)^\alpha) \qquad (7)$$

where the cutoff J_n is defined as the smallest positive integer such that

$$\sum_{0<|j|<J_n} |j|^{2l}|h_K(j)|^{-2}((J_n/|j|)^\alpha - 1) > nQ \ . \qquad (8)$$

The following sequence r_n plays a role of the indicator which shows when sharp optimal estimation is possible. Set

$$r_n = \min_{|j|<J_n}\{\delta_n^2(\alpha, Q, l)/|j^l h_K^{-1}(j)|^2\} \ . \qquad (9)$$

We shall see that if $r_n \to \infty$ then sharp optimal estimation is possible and otherwise it is impossible. The underlying idea of the sequence r_n is as follows. If $r_n < C < \infty$ then, following the terminology of Donoho and Liu (1991) and Fan (1993), the difficulty of the nonparametric problem may be captured by the hardest one-dimensional subproblem. As a result there is no sharp lower bound because there is no sharp lower bound for a one-dimensional problem.

Define a real-valued estimate

$$\tilde{f}_n^{(l)}(t, J, \{\lambda_j\}) = (2\pi)^{-1} \sum_{|j|\le J} \lambda_j Y_j h_K^{-1}(j)(-ij)^l \exp(-ijt). \qquad (10)$$

The following assertion shows that this estimate has the property of optimal MISE convergence under appropriate choice of J and $\{\lambda_j\}$. Here we assume that $\lambda_j^* = 1 - (|j|/J_n)^\alpha$ and whenever the Lipschitz space is under consideration we set $Q = 1$.

Theorem 1 *Let $0 < |h_K(j)| < \infty$ and $l < \alpha$. Then the estimate (10) has the following optimal asymptotic (as $n \to \infty$) properties of MISE convergence:*

(i) If $r_n \to \infty$ then estimate $\tilde{f}_n^{(l)}(t) = \tilde{f}_n^{(l)}(t, J_n, \{\lambda_j^*\})$ has sharp optimal minimax MISE convergence over the Sobolev class $H(\alpha, Q)$ of functions f, that is,

$$\sup_{f\in H(\alpha,Q)} E_f\{\int_0^{2\pi} (\tilde{f}_n^{(l)}(t) - f^{(l)}(t))^2 dt\}$$

$$= \inf \sup_{f\in H(\alpha,Q)} E_f\{\int_0^{2\pi} (\hat{f}_n^{(l)}(t, \alpha, Q, h_K) - f^{(l)}(t))^2 dt\}$$

$$= (1 + o(1))\delta_n^2(\alpha, Q, l)$$

where the inf is over all possible estimates $\hat{f}_n^{(l)}(t, \alpha, Q, h_K)$.

(ii) Estimate $\hat{f}_n^{(l)}(t) = \tilde{f}_n^{(l)}(t, J_n, \{1\})$ has rate optimal MISE convergence over either Sobolev $H(\alpha, Q)$ or Lipschitz $Lip(\alpha)$ classes of estimated functions and

$$\sup E_f\{\int_0^1 (\hat{f}_n^{(l)}(t) - f^{(l)}(t))^2 dt\} = O(1)\delta_n^2$$

where the sup *is over either the Sobolev $H(\alpha, Q)$ or Lipschitz $Lip(\alpha)$ classes of estimated functions f.*

We see that the smoothing coefficients $\{\lambda_j\}$ have been employed only to obtain sharp optimal MISE convergence, that is, the best constant and rate of MISE convergence. They reflect the statistical nature of the problem rather than approximation of a function via a trigonometric polinom.

The situation drastically changes when L_p-norms with $p \neq 2$ are used to measure the accuracy of fitting. Unfortunately, straightforward implementation of the Fourier approximation gives the optimal fitting only within the logarithmic factor, see more in Butzer and Nessel (1971)). However, implementing of a smoothing allows us to avoid this decreasing in accuracy of approximation. For arbitrary p the familiar de La Vallée Poussin sum is a good alternative and the corresponding estimate is defined as $\tilde{f}_n^{(l)}(t, 2J, \{\mu(j, J)\})$ with $\mu(j, J) = 1$ if $|j| \le J$, $\mu(j, J) = 2 - |j|/J$ if $J < |j| < 2J$ and $\mu(j, J) = 0$ otherwise.

Note that this sum gives an excellent approximation to the considered functions $f(t)$. In fact, de La Vallée Poussin sums are within a constant factor 4 of the best sup-norm approximation by trigonometric polynomials of a given order. See more about estimates based on this sum in Ibragimov and Khasminskii (1981) and Efromovich and Low (1994).

The interested reader is referred to Efromovich (1994) where risks in L_p-norms, $1 \le p \le \infty$, are investigated. It is shown that the sequence $\delta_n = \sqrt{\delta_n^2}$ defines both sharp optimal and rate optimal risk convergence in different L_p-norms for estimates of $f^{(l)}$ whenever $1 \le p < \infty$, moreover, this is the optimal rate for $p = \infty$ as well whenever h_k corresponds to the supersmooth case.

3 Density Deconvolution for Irregular Case

Consider the discussed in Introduction problem of density deconvolution when $h_\epsilon(j) = 0$ for some integers j but (4) holds.

Let s_{jn} be a sequence in n and j such that $s_{-jn} = -s_{jn}$ and for each $j \ge 0$ and n the sequence minimizes up to a constant factor the error $e(s, j, n) = |s|^2 + n^{-1}|h_\epsilon(j+s)|^{-2}$ over $|s|^2 \le Cn^{-1}|h_\epsilon(j+s)|^{-2}$. Set $R(J, n) = [n^{-1}J\sum_{|j|\le J}|h_k(j + s_{jn}|^{-2} + J^{-2\alpha}]^{1/2}$. The sequence $R(J, n)$ is the upper bound (up to a constant factor) for risk of the recommended orthogonal series estimate with the cutoff J. Then we define optimal sequence J_n as an increasing sequence of positive integers which minimizes rate of decaying $R(J, n)$ as $n \to \infty$.

We are now in a position to define a consistent real -valued orthogonal series estimate of the l-th derivative of density f as

$$\hat{f}_n^{(l)}(x) = (2\pi)^{-1} \sum_{|j|<2J_n} (-ij)^l$$

$$\times \mu(j, J_n)\hat{h}_X(j+s_{jn})h_\epsilon^{-1}(j+s_{jn})\exp(-ijx) \qquad (11)$$

where $\mu(j, J_n) = 1$ if $|j| \le J_n$ and $\mu(j, J_n) = 2 - |j|/J_n$ if $J_n < |j| < 2J_n$ are the discussed above smoothing coefficients of de La Vallée Poussin sum.

The reader who is primarily interested in a traditional MISE may simplify the estimate and consider $\tilde{f}_n^{(l)}$ which is defined by (11) with $\mu(j, J) \equiv 1$.

Theorem 2 *Suppose that (4) holds,* $1 \le p \le \infty$, $x_0 \in [0, 2\pi)$ *and* $l < \alpha$. *Then:*

(i) Estimate (11) is consistent and

$$\sup_{f \in Lip_\alpha} E_f\{\|\hat{f}_n^{(l)} - f^{(l)}\|_p\} < CJ_n^l R(J_n, n)$$

where $J_n^l R(J_n, n) \to 0$ *as* $n \to \infty$.

(ii) If distribution of ϵ *is supersmooth then the estimate (11) is rate optimal in sense of the lower bounds of Fan (1991,1993), namely,*

$$\sup_{f \in Lip_\alpha} E_f\{\|\hat{f}_n^{(l)} - f^{(l)}\|_p\} = O((\ln(n))^{-(\alpha-l)/\beta}) \ ,$$

$$\sup_{f \in Lip_\alpha} E_f\{|\hat{f}_n^{(l)}(x_0) - f^{(l)}(x_0)|^2\} = O((\ln(n))^{-2(\alpha-l)/\beta}).$$

(iii) For $p = 2$ *statements (i) and (ii) also hold for the simplified estimate* $\tilde{f}_n^{(l)}$ *and under assumption of part (ii) MISE of this estimate decreases as* $O((\ln(n))^{-2(\alpha-l)/\beta})$. *for all* $f \in Lip(\alpha)$

Several remarks are to be made. In the first place, in contrary to the blurred image reconstruction model of Korostelev and Tsybakov (1994), for the considered setting irregularity does not necessarily implies inconsistency. Secondly, for the supersmooth case there is no influence of p, i.e. the loss function, on risk convergence, recall that this is not the case for direct observations (see Ibragimov and Khasminskii (1981)). Thirdly, slightly modified procedure allows to construct rate-optimal procedure for the ordinary smooth case as well, see Efromovich (1994). Finally, the procedure (11) may be recommended for the practically important case of small samples whenever $|h_\epsilon(j)|$ takes on relatively small values.

The following examples clarify the issue of the irregularity for density deconvolution model.

Example 1. In this example we analyze some familiar measurement errors which may lead to irregular setting. Let ϵ be uniformly distributed over interval (a, b) then $h_\epsilon(v) = \exp(iav)[\exp(i(b-a)v) - 1]/(i(b-a)v)$ (see Feller (1966)). The irregularity occurs whenever $(b-a)j = 2\pi r$ for some integers j and r. However, for this familiar measurement error consistent estimation is always possible because (4) holds. For if $h_\epsilon(j) = 0$ then it is elementary to verify that for some positive constants C_1, C_2 and $|s| < (\pi/2)/(b-a)$ the relations $C_1|s||j|^{-1} \le |h_\epsilon(j+s)| \le C_2|s||j|^{-1}$ hold.

Recall that $s_{-jn} = -s_{jn}$. To find s_{jn} for $j \ge 0$ let κ_j be such that $|\kappa_j| \le \pi$ and $(b-a)j = 2\pi r + \kappa_j$ for some integer r. Then one can set $s_{jn} = 0$ if $|\kappa_j| > n^{-1/4}|j|^{1/2}$ and $s_{jn} = n^{-1/4}|j|^{1/2}\mathrm{sgn}(\kappa_j)$ otherwise. Notice that even if $\kappa_j \ne 0$, i.e. for regular case, it may be worthwhile to implement our method and to estimate $h_f(j + s_{jn})$ rather than $h_f(j)$.

Interesting situation occurs when ϵ is a discrete random variable, that is, it takes on values a_r with nonzero probability p_r. Particularly, if $a_1 = 0$, $a_2 = \pi$ and $p_1 = p_2 = 1/2$ then $h_\epsilon(j) = 0$ for odd j.

Example 2. A wide class of "irregular" measurement errors can be generated by mixing the random variables described in Example 1 and traditionally studied "regular" measurement errors whose characteristic functions do not vanish.

Consider a mixture $\epsilon = \epsilon_1 + \epsilon_2$ of two random variables where ϵ_1 is any random variable from Example 1 with the characteristic function $h_{\epsilon_1}(v)$ which vanishes when v is equal to some integers and ϵ_2 is a random variable whose characteristic function $h_{\epsilon_2}(v)$ does not vanish. Then irregularity always occurs because $h_\epsilon(v) = h_{\epsilon_1}(v)h_{\epsilon_2}(v)$.

Such modelling a measurement error is very convenient for Monte Carlo simulations. For instance, to model an irregular supersmooth setting one chooses ϵ_2 from the list of the supersmooth random variables (remind that it includes all non-degenerated stable random variables and their mixtures) and then adds any random variable ϵ_1 discussed in Example 1.

Example 3. Interesting situation occurs if random variable X is projected onto a circle with unit radius and we observe this projection rather than X, that is, we observe $X' = X - \lfloor X/2\pi \rfloor$ instead of X. It is plain to see that for regular setting with $s_{jn} = 0$ this reduction of information does not effect our estimate (2.2).

Situation changes for irregular setting when this projecting does effect our estimate (11). However, it is not difficult to verify that whenever $E_f\{|\exp(is(X' - X)) - 1|^2\} < C|s|^2$, for instance the latter is the case when X has a finite second moment, then similarly to the regular case this reduction of information does not change the assertion of Theorem 2.1. Moreover, the procedure of Efromovich (1994) is not so sensitive to such projection.

Example 4. Recall that for nonparametric regression deconvolution Korostelev and Tsybakov (1993) have explored an example when vanishing of $h_K(j)$ implies inconsistency. Is there similar setting for the density deconvolution?

To implement the underlying idea of their example we are to suppose that both f and k are periodic with period 2π over all reals, however there is no periodic densities since any density is to be integrated to one.

Therefore, we are to change our setting. Let the random variable X be distributed on a circle with unit radius according to the density $g(x) = \int_0^{2\pi} f(t)k(t-x)dx$ where $k(x)$ is a known periodic function and $f(x)$ is estimated periodic function. For this mathematical model we may conclude that there is no consistent estimator whenever $h_k(j) = 0$ for at least one j.

The last circular model is a *very* special one and it sheds light on the circumstances when irregularity implies inconsistency, see also Hall (1990).

4 Extensions

The interested reader can easily extend the obtained results to the different models. The only model, which requires some explanation, is the nonparametric regression with errors in predictors.

Recall that for this model the unobserved predictor U is a random variable with density $p(u) > 0$. The recommended procedure of estimation is as follows. First, we use the estimate (3) where the empirical Fourier transform $n^{-1}\sum_{l=1}^{n} Y_l e^{ijX_l}$ is used instead of the empirical characteristic function. Notate the obtained estimate as $\hat{\psi}_n(u)$. It is not difficult to verify that $\hat{\psi}(u)$ is an estimate for the ratio $f(u)/p(u)$. Thus, if $p(u)$ is known then one can set $\hat{f}_n(u) = \hat{\psi}_n(u)p(u)$, otherwise estimate of $p(u)$ discussed in Section 3 can be plugged-in.

An interesting possible extension is a construction of an adaptive procedure. There are two different kinds of adaptation. The first one is to adapt to unknown smoothness of the underlying function f. The second one is a data-driven procedure of estimation f when h_K, the kernel of convolution, is unknown.

5 Conclusion

We have explored a problem of nonparametric curve estimation for convolution model. The proposed orthogonal series estimator has asymptotically sharp-optimal property of MISE convergence as well as rate-optimal risk convergence for a wide variety of loss functions. The procedure allows to treat both regular and irregular cases.

An interesting feature of this estimator is that it is similar to the estimators based on direct observations. The estimator may be used for density, filtering and nonparametric regression deconvolution as well as for nonparametric regression with errors in predictors.

6 References

Brown, L. and Low, M. (1992). Asymptotic equivalence of nonparametric regression and white noise. *Technical Report* Cornell Univ.

Butzer, P. and Nessel, R. (1971). *Fourier Analysis and Approximation.* Birkhauser Verlag Basel.

Donoho, D. and Low, M. (1992). Renormalization exponents and optimal pointwise rates of convergence. *Ann. Statistics* **20**, 944-970.

Donoho, D. and Liu, R. (1991). Geometrizing rate of convergence III. *Ann. Statistics* **19**, 668-701.

Efromovich, S. (1985). Nonparametric estimation of a density with unknown smoothness. *Theory Probab. Applications* **30** 557-568.

Efromovich, S. (1992). On orthogonal series estimators for random design nonparametric regression. *Computing Science and Statistics.* **24** 375-379.

Efromovich, S. (1994). Optimal deconvolution. *Technical Report.* University of New Mexico.

Efromovich, S. and Low, M. (1994). Adaptive estimates of linear functionals. *Probability Theory and Related Fields* **98** 261-275.

Fan, J. (1991). On the optimal rates of convergence for nonparametric deconvolution problems. *Ann. Statistics* **19**, 1257-1272.

Fan, J. (1993). Adaptively local one-dimensional subproblems with application to a deconvolution problem. *Ann. Statistics* **21**, 600-610.

Fan, J. and Truong. Y. (1993). Nonparametric regression with errors in variables. *Ann. Statistics* **21**, 1900-1925.

Feller, W. (1966). *An Introduction to Probability Theory and its Applications* Chapman&Hall.

Hall, P. (1990). Optimal convergence rates in signal recovery. *Ann. Probability* **18** 887-900.

Ibragimov,I. and Khasminskii, R. (1981). *Statistical Estimation: Asymptotic Theory.* Springer.

Korostelev, A. and Tsybakov, A. (1993). *Minimax Theory of Image Reconstruction.* Springer-Verlag.

Open Questions in the Application of Smoothing Methods to Finite Population Inference

Alan H. Dorfman
Office of Survey Methods Research
Bureau of Labor Statistics
2 Massachuesetts Avenue, N.E.
Washington D.C. 20212

Survey sampling, or finite population estimation, has been a domain unto itself, with theory and methods distinct from mainstream statistics. Nonparametric regression applied to survey results can yield more efficient estimation of population quantities than standard methods of survey inference, and, conceptually, bridges the gap between traditional mainstream statistics and survey sampling. The application of nonparametric regression to finite population estimation raises new questions for survey sampling and for the field of nonparametric regression..

1. INTRODUCTION

In this paper, we consider the application of nonparametric regression to the estimation of finite population "parameters" based on a sample from the population. Given a population *P* of N units for each of which there is a variable Y of interest, with values available on a sample *s* of *P*, we wish to estimate the population total $T=\sum_P Y_i$. We assume that an auxiliary variable *x* related to *Y* is available for the entire population.

Typically (although not always) the sample is selected according to a probability design, and the probabilities that an item is included in the sample is incorporated into the estimator. For example, stratifying on the auxiliary, and using stratified random sampling without replacement leads to the "expansion estimator" $\hat{T}_{\exp}=\sum_h \pi_h^{-1}\sum_{s_h} Y_{hi}$, where, for *h*=1,2,...,H, π_h are the probabilities of including unit Y_{hi} in the sample component s_h of the *h*th stratum.

Here we suggest a new estimator which uses nonparametric regression and is based on the prediction approach to survey inference (for example, see Royall and Herson, 1973). Related work in the application of nonparametric regression to sampling may be found in (Cheng 1994, Dorfman and Hall 1992, Jones and Bradbury 1993, Kuk 1993).

2. A NEW ESTIMATOR OF TOTAL

Consider the model

$$Y_i=m(x_i)+\sigma(x_i)e_i, i=1,..,N \qquad (1)$$

with $m(\cdot)$ a smooth function and the e_i independent with mean 0 and constant variance. Let $K(u)$ be a symmetric density function, for example the standard normal density function. For a chosen scaling factor ("bandwidth") *b*, define $K_b(u)=b^{-1}K(u/b)$, and weights $w_i(x)=K_b(x_i-x)\Big/\sum_{i=1}^{n}K_b(x_i-x)$. We consider the Nadaraya-Watson estimator of $m(x)$ given by

$$\hat{m}(x)=\sum_i w_i(x)Y_i. \qquad (2)$$

Under reasonable conditions on $m(x)$ and the design points *x*, $\hat{m}(x)$ is consistent for $m(x)$, as $b\to 0$, $nb\to\infty$.

If we let $x=x_j$, the values of *x* in the part of the population which has *not* been sampled, then a natural estimate of *T* is

$$\hat{T}_{np}=\sum_s Y_i+\sum_{P-s}\hat{m}(x_j)$$

As with prediction-based estimators generally, this estimator ignores sampling probabilities.

The conditional mean and variance of $\hat{T}_{np}-T$ under (1) are readily expressed as

$$E(\hat{T}_{np}-T|X_P)=\sum_{j\in P-s}\hat{d}_s(x_j)^{-1}(nb)^{-1}$$
$$*\sum_{i\in s}\left[K\{(x_j-x_i)/b\}\{m(x_i)-m(x_j)\}\right] \qquad (3)$$

and

$$\operatorname{var}(\hat{T}_{np}-T|X_P)=\sum w_i^2\sigma^2(x_i)+\sum\sigma^2(x_j), \qquad (4)$$

where $\hat{d}_s(x_j)=(nb)^{-1}\sum_{i\in s}K\{(x_i-x_j)/b\}$, $w_i=\sum_j(nb)^{-1}K\{(x_i-x_j)/b\}\{\hat{d}_s(x_j)\}^{-1}$, and X_P is the population vector of *x*-values. Note that $\hat{d}_s(x_j)$ is the standard Nadaraya-Watson estimator of $d_s(x_j)$. We have the following theorem along lines suggested in (Chambers, Dorfman, and Hall 1992), (Dorfman and Hall 1992), and (Ruppert and Wand 1993):

Theorem 1. Let $K(u)$ be a symmetric density function with $\int uK(u)du=0$ and $k_2\equiv\int u^2K(u)du>0$; assume *n* and *N* increase together such that $n/N\to\pi$, with $0<\pi<1$; assume sample and non-sample values of *x* are in the interval $[c,d]$ and are generated by densities d_s and

d_{P-s} respectively, both bounded away from zero on $[c,d]$, and assumed to have continuous first derivatives; let $\hat{m}(x)$ be defined as at (2) above; assume $m(x)$ has a continuous second derivative, and let $\beta(x)=d_s(x)m''(x)+2d_s'(x)m'(x)$; then

$$E\left(\hat{T}_{np}-T|X_P\right)=b^2(N-n)(k_2/2)*$$
$$\int\beta(x)d_s(x)^{-1}d_{P-s}(x)dx$$
$$+O_p\left(nb^3+n^{1/2}b^{1/2}\right) \qquad (5)$$

and

$$\mathrm{var}\left(\hat{T}_{np}-T|X_P\right)=(N-n)^2n^{-1}$$
$$*\int\sigma^2(x)d_s(x)^{-1}\left[d_{P-s}(x)\right]^2dx$$
$$+(N-n)n^{-1}b^{-1}\int K^2(u)du*\int\sigma^2(x)d_s^{-1}(x)d_{P-s}(x)dx$$
$$+(N-n)^2n^{-1}b^2k_2^2\int c^*(x)d_s(x)dx$$
$$+(N-n)\int\sigma^2(x)d_{P-s}(x)dx+O_p\left(nb^3+n^{1/2}b^{-1/2}\right) \quad (6)$$

We leave unspecified $c*(x)$, a complicated function of the derivatives of $d_s(x)$ and $d_{P-s}(x)$. **Proof of Theorem 1.** The basic mechanism driving the proofs is that if, for any expression *Z*, *E(Z|u)=A(u)+O(B)*, and *var(Z|u)=O(C)*, then $Z=A(u)+O_p\left(B+C^{1/2}\right)$, a result that follows from the Chebychev inequality. In the following remarks, *i*, *i'* etc. index sample units, and *j*, *j'* nonsample units.

Transition from (3) to (5): Note that $\hat{d}_s(x_j)=d_s(x_j)+b^2d_s''(x_j),+O_p\left(b^3+n^{-1/2}b^{-1/2}\right)$ a result that follows directly from calculation of the mean and variance of $\hat{d}_s(x_j)$. In similar fashion, conditional on x_j,

$$(nb)^{-1}\sum_{i\in s}K\left(\left[x_i-x_j\right]/b\right)\left\{m(x_i)-m(x_j)\right\}=$$
$$b^2d_s''(x_j)k_2/2+O_p\left(b^3+n^{-1/2}b^{1/2}\right),$$

since the left-hand expression has mean $b^2d_s''(x_j)k_2/2+O(b^3)$ and variance

$$n^{-1}b^{-2}\{E\left(K^2\left(\left[x_i-x_j\right]/b\right)\left[m(x_i)-m(x_j)\right]^2|x_j\right)$$
$$-E^2\left(K\left(\left[x_i-x_j\right]/b\right)\left[m(x_i)-m(x_j)\right]|x_j\right)\}=O(n^{-1}b);$$

the last equality follows from

$$E\left(K^2\left(\left[x_i-x_j\right]/b\right)\left[m(x_i)-m(x_j)\right]^2|x_j\right)=$$
$$\int K^2\left(\left[w-x_j\right]/b\right)\left[m(w)-m(x_j)\right]^2d_s(w)dw=$$
$$b\int K^2(u)\left[ubm'(x_j)+O(u^2b^2)\right]^2$$
$$*\left[d_s(x_j)+ubd_s'(x_j)+O(u^2b^2)\right]du$$

Combining the *(N-n)* terms and repeating the argument leads to (5).

Transition from (4) to (6): Let *M=N-n*. The second term of (4) is straightforward to deal with. The main task is developing an expression for w_i^2. We have

$$E\left(w_i^2|x_i\right)=Mn^{-2}b^{-2}$$
$$*E\left\{K^2\left(\left[x_j-x_i\right]/b\right)d_s^{-2}(x_j)\left(1+O_p\left[b^2+n^{-1/2}b^{-1/2}\right]\right)|x_i\right\}$$
$$+M(M-1)n^{-2}b^{-2}$$
$$M(M-1)n^{-2}b^{-2}E^2\{K\left(\left[x_j-x_i\right]/b\right)d_s^{-1}(x_j)$$
$$*\left(1+c(x_j)b^2+O_p\left(b^3+n^{-1/2}b^{-1/2}\right)\right)|x_i\}$$

By the usual Taylor expansion, the first term equals $Mn^{-2}b^{-1}\int K^2(u)dud_s^{-2}(x_i)d_{P-s}(x_i)$ $+O\left(n^{-1}+n^{-1.5}b^{-2.5}\right)$; the second term equals $M^2n^{-2}\left\{d_s^{-2}(x_i)d_{P-s}^2(x_i)+k_2c(x_i)b^2\right\}+$ $O\left(n^{-1}+b^3+n^{-1/2}b^{-1/2}\right)$. Further, in $\mathrm{var}\left(w^2(x_i)|x_i\right)=E\left(w^4(x_i)|x_i\right)-E^2\left(w^2(x_i)|x_i\right)$, the dominant terms of these two terms cancel, and we find $\mathrm{var}\left(w^2(x_i)|x_i\right)=O\left(n^{-1}b^{-1}\right)$. Combining expressions yields

$$w_i^2=M^2n^{-2}\left\{d_s^{-2}(x_i)d_{P-s}^2(x_i)+k_2c(x_i)b^2\right\}+$$
$$Mn^{-2}b^{-1}\int K^2(u)dud_s^{-2}(x_i)d_{P-s}(x_i)+$$
$$O_p\left(n^{-1}+b^3+n^{-1/2}b^{-1/2}\right).$$

Summing over *i* yields (6).♦

We note the following consequences:

(i) The conditional relative bias is $O_p\left(b^3+n^{-1/2}b^{1/2}\right)$; this goes to zero so long as $b\to0$. (ii) The variance is $O_p(n)$ so long as the weak conditions $b\to0$, $nb\to\infty$ are met. (iii) If $b=Cn^{\varepsilon}$ for $\varepsilon<-1/4$, then the ratio $E\left(\hat{T}_{np}-T|X_P\right)/\mathrm{var}^{1/2}\left(\hat{T}_{np}-T|X_P\right)$ is asymptotically zero in probability, a next-best-to-unbiasedness condition that allows for constructing confidence intervals for *T* based on estimates of variance; we note that the standard bandwidth $b=Cn^{-1/5}$, optimal under mean square error criteria for *m(x)* itself, is too large for the bias to become negligible, so that other than standard methods of selecting bandwidth seem to be in order. (iv) Under simple random sampling, or more generally when $d_s(x)=d_{P-s}(x)$, the variances of the simple random sample based $\hat{T}_{\exp}$ and $\hat{T}_{np}$ are equal (to first order), but the bias of $\hat{T}_{\exp}$ is of the same order $O_p\left(n^{1/2}\right)$ as the root of the variance, unless *m(x)* is a constant on [*c*,*d*]; hence $\hat{T}_{\exp}$ lacks the desirable property mentioned in the previous remark. In the case of stratified random sampling, where a finite number of strata grow without limit, the bias of $\hat{T}_{\exp}$ is likewise in general $O_p\left(n^{1/2}\right)$ unless *m(x)* is constant on each stratum. (v) The results on the bias of $\hat{T}_{np}$ hold whether or not the sample and non-sample densities are the same; this suggests that *balance* (Royall and Herson 1973) plays a less important role with this estimator; however, we cannot be indifferent to the spread of the *x*'s, since the efficiency of nonparametric regression can be affected; compare (Chu and Marron 1991). (vi) In the variance the implicit term $O_p\left(n^{1/2}b^{-1/2}\right)$ is of larger order than the explicit $O\left(b^{-1}\right)$ term for $b = Cn^{\varepsilon}$, $\varepsilon > -1$; this suggests that plug in methods for estimating bandwidth based on (4) and (6) will be ineffective. (vii) The condition of the theorem that *n* is of the same order as *N* can be loosened to *n=O(N)*, at the price of complicating the expression of the $O_p(\ )$ terms in (4) and (6).

To the end of estimating variance, we follow a suggestion of Rose (1978) and define

$$\hat{\sigma}^2(x) = \hat{m}_{2l}(x) - \hat{m}_h(x), \qquad (7)$$

where $\hat{m}_h(x)$ is a pilot estimator of *m(x)* based on bandwidth *h*, as in (2), and $m_{2l}(x) = (nl)^{-1}\sum_s K\left(\left[x_i - x\right]/l\right)Y_i^2 \Big/ \hat{d}_{sl}(x)$ is a non-parametric regression estimator of $m_2(x) \equiv E\left(Y_i^2 | x\right)$ based on a possibly different bandwidth *l*. We allow *h* and *l* to be different.

Theorem 2. Let

$\hat{\mathrm{var}}\left(\hat{T}_{np} - T | X_P\right) = \sum w_i^2 \hat{\sigma}^2(x_i) + \sum \hat{\sigma}^2(x_j)$with $\hat{\sigma}^2(x)$ as defined in (7). Then $\hat{\mathrm{var}}\left(\hat{T}_{np} - T | X_P\right)$-

$$\mathrm{var}\left(\hat{T}_{np} - T | X_P\right) =$$

$$O\left(nl^2\right) + O_p(nl^3 + n^{1/2}\left[l^{1/2} + (nl)^{-1/2} + 1\right].$$

$$+n\left[(nh)^{-1/2} + h^2\right] + 1)$$

3. EMPIRICAL RESULTS

We consider a population consisting of N=400 establishments. The data is taken from the United States Bureau of Labor Statistics' 1991 Occupational Compensation Survey. The variable of interest Y is the total wages paid to workers in a selected group of occupations; x is the total number of workers in each establishment including those in occupations outside the selected group. From this population, 100 samples were taken, using stratified random sampling without replacement; for *h*=1,2,3 , n_h=20 points were taken from each of three strata of sizes N_h= 202, 114, and 84 respectively. Three classes of company size, viz. $0 < x < 250$, $250 \le x < 1000$, and $1000 \le x$, determined the strata.

For each sample, we calculated (i) the nonparametric regression-based estimator; (ii) several design-based estimators of the total, namely the expansion estimator, and the combined and separate ratio and regression estimators (Cochran 1977); and poststratified estimators and (iii) the linear-model based estimator with different assumed variance structures. The auxiliary variable was log-transformed for the nonparametric regression-based estimator, and for the design-based estimators in some instances. Three bandwidths were used which were judged to give reasonable results based on visual inspection of fits on a single sample, in two ways, namely, for immediate use in the nonparametric estimator of total $\hat{T}_{np}$, or as seeds to choose the bandwidthin by the algorithm of Hardle, Hall, and Marron (1992).

Table 1 gives summary results in the form of the average relative error $\sum_{r=1}^{100} T^{-1}\left(\hat{T}_r - T\right)\Big/100$ and the root average squared error $\left\{\sum_{r=1}^{100}\left(\hat{T}_r - T\right)^2 \Big/ 100\right\}^{1/2}$ (RASE), where $\hat{T}_r$ is one of the estimators of *T* computed for sample *r*. In terms of the RASE, we note that the combined and separate regression estimators do not much improve the expansion estimator, and in fact do a lot worse unless the auxiliary variable is log-transformed. The nonparametric regression-based estimator is more efficient (i.e. has smaller average squared error) than the best of the design-based estimators, at the two larger bandwidths. It has about the same efficiency as the expansion estimator at the smaller bandwidth. The bandwidth selection procedure does not do as well as the naked eye.

Greatest efficiency was achieved by the model-based estimator relying on a linear model, with variance assumed proportional to x^2, but there is a drop in efficiency with the other variance structures well below the nonparametric estimator at larger bandwidth. Note that the nonparametric regression estimator does not require us to specify the variance structure. Table 2 gives results on variance estimation for the expansion, poststratified, and nonparametric regression estimators. The mean root of the variance estimates tends to be lower than the RASE, especially for the nonparametric regression estimator, and coverage is low. For the stratification estimator with finest stratification, the variance estimator was available in only 82% of runs. We note the anomalous behavior of the nonparametric regression variance estimator, which tends to get smaller at smaller bandwidths, when the RASE is largest. Allowing *l*, the bandwidth used to estimate the $m_2(x)$ component of $\sigma^2(x)$, to be chosen independently worked to moderate advantage here, somewhat increasing the average root variance estimates, and the coverage. In one run for seed(*b*)= 0.25, the variance estimate was negative. In practice, one could have recourse to forcing *l*=*h*, in such a case.

4. QUESTIONS

The empirical results suggest that the nonparametric regression-based estimator of a finite population total is a strong rival to established estimators. It has the quality of automaticity we associate with design-based estimators, but is likely to reflect better the actual structure of the data, yielding greater efficiency. It can be costly in computer power, and may not do as well as a parametric-model based estimator, when the modelling process is done carefully on well-behaved data.

Further research on the nonparametric regression-based estimator is needed:

* Automatic bandwidth selection. Standard bandwidth selection methods such as that of Hardle, Hall, and Marron (1992) which we used in the simulation study aim at estimating a bandwidth that minimizes the average square error of the $m(x_i)$, $i \in s$. This bandwidth has the property that $h = Cn^{-1/5}$, outside the range of acceptable bandwidths in note (ii) of Section 2. It is in fact larger, so that curves based on standard methods will tend to be too smooth for deriving the estimate of total. One is tempted to use plug-in methods that would minimize the MSE of $\hat{T}_{np}$, but as in note (vi) of Section 2, there seem to be intrinsic barriers to this approach.

* Variance estimation. The results of the simulation suggest a need for further work here that would give better coverage, and estimated standard deviations with expectation closer to root mean sauare error of $\hat{T}_{np}$.

* Can we improve $\hat{T}_{np}$ by alternatives to straightforward Nadaraya-Watson? Many methods deserve serious consideration, including adaptive bandwidth and local linear regression (Fan and Gibjels 1992). Possibly estimates or a priori guesses of the variance structure could profitably be incorporated into the nonparametric regression based estimator.

* As noted in section 2 note (v), the sample design is much less of a concern when we use nonparametric regression than in strictly model-based estimators. But because of the dangers of extrapolation and "internal extrapolation" (dealing with holes), this cannot mean that any sample is permissible; what are the boundaries of the permissible and also what characterizes samples with smallest MSE?

REFERENCES

Chambers, R. L., Dorfman, A. H., and Hall, P. (1992), Properties of Estimators of the Finite Distribution Function, *Biometrika* 79, 577-582.

Cheng, P. E. (1993), Nonparametric Estimation of Mean Functionals with Data Missing at Random, *J. Am Statist. Assoc.* 89, 81-88.

Chu, C. K. and Marron, J. S. (1991), Choosing a Kernel Regression Estimator (with commentary), *Statistical Science* 6, 404-436.

Cochran, W. G. (1977), *Sampling Techniques*(3rd ed.), Chichester: John Wiley.

Dorfman, A. H. and Hall, P. (1992) Estimators of the finite population distribution function using nonparametric regression, *Annals of Statistics*, 21, 1452-1475.

Fan and Gibjels (1992), Variable Bandwidth and Local Linear Regression Smoothers, *Annals of Statistics*, 20, 2008-2036.

Hardle, W., Hall, P. and Marron, J. S. (1992), Regression Smoothing Parameters that are not far from their Minimum, *J. Am Statist. Assoc.* 87, 227-233.

Jones, M. C. and and Bradbury, I. S. (1993) Kernel Smoothing for Finite Populations *Statistics and Computing* 3, 45-50.

Kuk, A. (1993) A Kernel Method for Estimating Finite Population Distribution Functions using Auxiliary Information, *Biometrika*, 80, 385-392.

Rose, R. L. (1978) *Nonparametric Estimation of Weights in Least-Squares Regression Analysis,* Ph. D. Thesis.

Royall, R. M., and Cumberland, W. G. (1981), An empirical study of the ratio estimator and estimators of its variance, *J. Am Statist. Assoc.* 76, 66-77.

Royall, R. M. and Herson, J. (1973) Robust Estimation in Finite Populations I, *J. Am Statist. Assoc.* 68, 880-893.

Ruppert, D. and Wand, M. P. (1993) Multivariate Locally Weighted Least Squares Regression, Preprint.

Table 1. Summary Statistics for Estimators of Total in Wage Population

Estimator	Average Relative Error	Root Average Squared Error/10^6	RASE$(\hat{T})$/ RASE$(\hat{T}_{exp})$
stratified expansion	0.035[a]	6.34[b] (.42)	1.00
poststratified 6 strata	0.033	6.36 (.51)	1.00
9 strata	0.023	6.07 (.53)	0.96
12 strata	0.15[d]	6.43[d] (.56)	1.01
combined ratio	0.040	6.22 (.56)	0.98
separate ratio	0.042	6.32 (.58)	1.00
combined regression	0.070	7.56 (.79)	1.19
combined regression(log)	0.033	6.16 (.40)	0.97
separate regression	0.069	7.71 (.80)	1.22
separate regression (log)	0.032	6.33 (.56)	1.00
linear model			
$\sigma^2(x_i) \propto x_i^0$	0.102	6.72 (.39)	1.06
$\sigma^2(x_i) \propto x_i$	0.067	6.94 (.62)	1.10
$\sigma^2(x_i) \propto x_i^2$	-0.063	4.56[c] (.33)	0.72
nonparametric regression			
b=0.25	0.040	6.50 (.59)	1.02
b=0.50	0.013	5.67[c] (.42)	0.89
b=0.75	0.001	5.40[c] (.38)	0.85
seed(*b*)=0.25	0.042	6.58 (.60)	1.04
seed(*b*)=0.50	0.025	6.13 (.54)	0.96
seed(*b*)=.75	0.018	5.90 (.50)	0.93

[a] Standard deviation for all entries is approximately 0.02. [b] Standard deviation is given in parentheses.

[c] The paired two sample *t*-test on the hypothesis $H\colon E\left\{\left(\hat{T}_{\bullet}-T\right)^2-\left(\hat{T}_{\exp}-T\right)^2\right\}=0$ is significant at $p=0.05$ [d]Based on 95 runs.

Table 2 Summary Statistics for Estimators of Variance of Total in Wage Population

Estimator	Root Average Squared Error/10^6	Average $\hat{v}^{1/2}/10^6$	Coverage~95% nominal	Average $\hat{v}^{1/2}/10^6$	Coverage~95% nominal
stratified expansion	6.34	6.63	91		
poststratified 6 strata	6.35	6.29	92		
9 strata	6.07	6.20[a]	88[a]		
12 strata	6.42[b]	5.95[c]	88 [c]		
nonparametric regression		*l*=*h*=*b*		*h*=*b*, seed(*l*)=*b*	
b=0.25	6.50	5.36	87	5.46	88
b=0.50	5.67	5.67	89	5.90	91
b=0.75	5.40	6.20	93	6.33	94
		l=*h*=*b*		*h*=*b*, seed(*l*)=seed(*b*)	
seed(*b*)=0.25	6.58	5.59	86	5.61 [d]	88 [d]
seed(*b*)=0.50	6.14	5.71	88	5.75	88
seed(*b*)=0.75	5.90	5.91	90	6.03	91

[a]Based on 97 runs. [b]Based on 95 runs. [c]Based on 82 runs. [d] Based on 99 runs.

Empirical Examination of an Efficient Robust Linear Regressor

Deborah Sturm
Department of Computer Science
College of Staten Island (CUNY)
2800 Victory Boulevard
Staten Island NY 10314
email: ddssi@cunyvm.cuny.edu

Abstract

Robust estimators are generally computationally intensive. A method which we call the inner products method (IP) is examined which is computationally comparable to Least Squares (LS) but is robust with respect to outliers. The algorithm consists of multiplying both the original model function and the interpolated data points by a set of "test functions" $\phi_1, \phi_2, \ldots, \phi_n$ and then integrating each to form the inner products. The two results are set equal to one another, yielding a system of constraints on the unknown parameters. This system of constraints is then used to solve for the unknown parameters. For linear models, the algorithm requires no initial estimate of the parameters, and the equations generated are always linear. With only Gaussian noise, least squares (LS) does slightly better than IP. When outliers are introduced IP does significantly better than LS and comparably with other robust methods such as *Least Median of Squares* (LMS) and *Iteratively Reweighted Least Squares* (IRLS). As with LMS, it can be used to predict model errors. Data from the literature, as well as simulated data, have been used to evaluate IP's performance. The algorithm generalizes to, and has been applied to, certain nonlinear models.

Introduction

Modeling of data arises in numerous applications in the natural and social sciences. Regression analysis, perhaps the most commonly used statistical technique, is used to fit observed data to a theoretical model function. While least squares methods are usually employed, they break down in the presence of non-Gaussian noise. Data sets often contain non-normally distributed noise including one or more wild observations (Rousseeuw 1987, Clancy 1947, Phillips 1983). Several robust methods have been suggested (Rousseeuw 1987, Huber 1981, Hampel 1986), but the algorithms are generally computationally intensive and often require initial guesses for the parameters, even in the linear case.

Background

Consider a set of n data points, (x_i, y_i), which are to be fitted to a model,

$$y(x) = y(x; a_1, \ldots, a_m), \quad (1)$$

where the a_j, $j = 1, \ldots, m$ are unknown parameters. The method of least-squares (LS) involves minimizing the function

$$\Phi(a_1, \ldots, a_m) = \sum_{i=1}^{n} [y(x_i; a_1, \ldots, a_m) - y_i]^2. \quad (2)$$

The least-squares estimator is a maximum likelihood estimator of the found parameters, if the errors in the data are independent and normally distributed with a constant standard deviation (Brownlee 1960). When the set of data points is thought to fit a linear combination of more than one function, one may use the generalized least squares (GLS) method. The least-squares method has been extended to the nonlinear setting using the Levenberg-Marquardt method (Bates 1988, Marquardt 1963, More 1977).

For linear problems several robust alternatives to LS estimation have been proposed which reduce the influence of outliers (Rousseeuw 1987, Hampel et. al. 1986). Among them are M-estimators (for a survey see Huber 1981). These estimates yield a system of equations which is typically nonlinear and difficult to solve. Iteratively reweighted least squares (IRLS) methods are then used. For a review see Holland 1977 and O'Leary 1990.

Rather than minimizing the *sum* of a function of the residuals, the least median of squares (LMS) method minimizes the *median* of the squares of the residuals (Rousseeuw 1987). LMS is robust with respect to outliers and leverage points but has a slow convergence rate.

Inner Product Method (IP)

Let $E(x)$ be an experimental data function whose domain is a finite set of points in the interval $[p, q]$. Let $M(x) \equiv M(x; a_1, a_2 \ldots, a_m)$ be the model function with unknown parameters $a_1, a_2, \ldots, a_m$ where x is defined in $[p, q]$. Ideally,

$$M(x) = E(x)$$

where both are defined. Thus we expect

$$\int_p^q M(x)\,dx = \int_p^q E(x)\,dx$$

where the data points are linearly interpolated. More generally, if $\phi_1, \phi_2, \ldots, \phi_m$ are arbitrary integrable functions on the interval $[p, q]$ (called *test functions*), we expect,

$$\int_p^q M(x)\phi_i(x)\,dx = \int_p^q E(x)\phi_i(x)\,dx, \quad i = 1, 2, \ldots, m. \tag{3}$$

Choosing good test functions, and integrating both sides of (3), we obtain a system with m equations and m unknowns.

We now solve for the unknowns to obtain the desired parameters. The motivation for this method is that the effect of the "random" noise in the data will be minimized after integrating against an integrable test function.

This algorithm, the *inner product method* (IP) (Sturm 1992), differs from other robust methods. To show this we define the following "universal" estimator system (U) where $\bar{\theta}$ is the vector of unknown parameters and $x_i = (x_{1i}, x_{i2}, \ldots, x_{in})^T$.

$$\sum_{i=1}^{n} \psi(y_i - \bar{\theta}\bar{x}_i - b) \begin{pmatrix} \phi_1(x_i) \\ \phi_2(x_i) \\ \vdots \\ \phi_m(x_i) \end{pmatrix} = 0. \tag{4}$$

Here ψ, ϕ_i are real valued functions of one variable with the property $\psi(0) = 0$. This last condition is imposed in order that the true values for the parameters are solutions to (4) when the data is perfect.

In matrix form for IP (4) becomes $A\theta = Y$, where $A =$

$$\begin{bmatrix} \sum x_{i1}\phi_1(x_i) & \sum x_{i2}\phi_1(x_i)\ldots & \sum \phi_1(x_i) \\ \sum x_{i1}\phi_2(x_i) & \sum x_{i2}\phi_2(x_i)\ldots & \sum \phi_2(x_i) \\ & \vdots & \\ \sum x_{i1}\phi_{m+1}(x_i) & \sum x_{i2}\phi_{m+1}(x_i)\ldots & \sum \phi_{m+1}(x_i) \end{bmatrix}$$

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \\ b \end{pmatrix}, \text{and,}$$

$$Y = \begin{bmatrix} \sum y_i\phi_1(x_i) \\ \sum y_i\phi_2(x_i) \\ \vdots \\ \sum y_i\phi_{m+1}(x_i) \end{bmatrix}$$

Note that LS for simple regression is obtained from U by specializing as follows: $\psi(x) = x, \phi_1(x) = x, \phi_2(x) = 1$. The IRLS method and the M-estimator method for simple regression are obtained from U by specializing as follows: $\phi_1(x) = x$ and $\phi_2(x) = 1$ and allowing ψ to be arbitrary. In our technique, the inner products method, we specialize $\psi(x) = x$ but allow ϕ_i to be arbitrary. A key advantage of IP is that for all linear model functions the resulting equations are linear. Therefore, there are no initial guesses for the parameters.

If we restrict the test functions to be the same as in LS, then the IP method can be restated as a minimization problem as follows. If we define the residual vector $r(\bar{\theta}) = A\bar{\theta} - Y$ then we can state the general data-fitting problem as the solution to

$$\min_{\bar{\theta}} f(\bar{\theta}) = \rho(r(\bar{\theta})). \tag{5}$$

In the case of least squares, the function ρ is

$$\rho(\bar{\theta}) = 1/2\sum_{i=1}^{n}(r_i(\bar{\theta})^2).$$

For the IP method, we define Z as a diagonal matrix with $(Z(i))$ along the diagonal, then the function $\rho(\bar{\theta}) = ||\bar{\theta}||_Z$. where the weighted norm $||x||_Z$ is defined as $x^T Z x$.

Augmented test functions

In order to find a better set of test functions than LS (i.e. one which produces a smaller error), we exploit the fact that the simple linear case satisfies the differential equation $f''(x) = 0$. Hence, if the data points (x_i, y_i) are thought to lie close to a straight line, then the discrete second derivative, $y_{i+1} - 2y_i + y_{i-1}$ should be close to zero. Thus if we set $D(i) = y_{i+1} - 2y_i + y_{i-1}$, then $D(i)$ measures how far y_i is from the model function. If $D(i)$ is big, then y_i is far from the line. Since $D(i)$ is not defined when i is an endpoint, we set $D(p)$ to $D(p+1)$ and $D(q) = D(q-1)$ where the data is defined on the interval $[p, q]$.

We modify the previous test functions ϕ using $D(i)$ to mollify the effect of the outliers. Let

$$Z(i) = \frac{1}{cD(i)^2 + 1}$$

where c is a relatively large positive number. Then $Z(i) \approx 1$ if y_i, y_{i-1} and y_{i+1} are "good" points, and $Z(i) \approx 0$ if y_i, y_{i-1} or y_{i+1} is an outlier. Now we replace $\phi(x)$ by $\phi(x)Z(x)$. Note that the farther the outlier is from the other data points, the less its influence will be.

However if several consecutive outliers happen to lie on a line, their influence will be exaggerated.

This method generalizes to higher dimensions. For purposes of explanation we will restrict our attention to two dimensions. If the data is equally spaced within a grid, we replace the discrete second derivative by the discrete Laplacian. In this case every interior point (x_{i1}, x_{i2}) has four immediate neighbors: $p_1 = (x_{i1}+1, x_{i2})$, $p_2 = (x_{i1}-1, x_{i2})$, $p_3 = (x_{i1}, x_{i2}+1)$, $p_4 = (x_{i1}, x_{i2}-1)$. The four vectors determined by these points, $v_k = p_k - p_0, k = 1 \ldots 4$, satisfy the following equation of linear dependence:

$$v_1 + v_2 + v_3 + v_4 = 0 \tag{6}$$

This equation of dependence is then used in order to adjust the data as shown above.

In higher dimensions, we will assume initially that the x_i are arranged in a cubical lattice. In other words, we shall assume that the x_i are the integral lattice points in a large cube. We shall say that two such points are "close neighbors" if their difference is a standard basis vector; in other words their difference equals $\pm(0, 0, \ldots, 0, 1, 0, \ldots, 0, 0)$. Thus each point has $2m$ close neighbors. Then if y_i is the response variable at the point x_i, the discrete Laplacian at x_i is the sum of the values at all the close neighbors minus $2m$ times the value at the center.

If the y_i all lie on a plane then the discrete Laplacian will be zero. Thus the discrete Laplacian measures how far the data is from being perfect. We note that an imperfect center value will create a much larger Laplacian than a neighboring value, since the coefficient of the center is $2m$ while the neighboring coefficients are all one. Note that this approach should improve with higher dimensions. For the boundary points which do not have $2m$ neighbors one can use fewer neighbors. At the corners, one might extrapolate or assume that the corners are outliers.

If the data is not equally spaced, we do not have the notion of immediate neighbors and a replacement for (6) is required. This replacement is obtained as follows: For every data point p_0, we divide the plane into three regions, by drawing three rays emanating from p_0 in such a way that the angle between any two is 120 degrees. Then choose one point p_i from the first region, p_2 from the second, and p_3 from the third (this will be possible for "most" points). Let $v_i = p_i - p_0, i = 1, 2, 3$ and solve the equation of dependence:

$$a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4 = 0.$$

The choice of p_i implies that all a_i may be taken to be positive. We normalize by taking $a_1^2 + a_2^2 + a_3^2 = 1$. This is used as a replacement for (6).

Numerical Results

Numerical results indicate that the IP method is more robust than least squares and compares favorably with other methods. The inner products method was examined using data from the literature. Figure 1 shows calibration data with least squares(LS), least median of squares (LMS), and inner products (IP) fits. The data is taken from (Massart 1986). The true relationship was $y = x$; note that only IP returns the exact slope.

Massart et. al. (Massart 1986) cite an application for robust estimation to calibration. They apply both LS and LMS to the same data set. If the two lines do not coincide, then LS is usually pulled by outliers at the end of the calibration range. As an example they study the calibration of lead measurements by plasma emission spectrometry. There are 13 data points, 10 of which are at the low end of the scale. These low concentration points determine the slope of the LMS and the IP line. The LS method fits the high end points. By comparing the two methods (Figure 2), the model error caused by curvature of the calibration line is revealed. Visual inspection alone would not have revealed this. Massart et. al. make the point that although this method of determining model errors is not a statistical test, there are no very good alternatives. Analysis of variance requires repeated measurements which may not be available. Residual analysis is affected by outliers in the least squares case. An F-test applied to this data did not show that a second-order model would be significantly better. Therefore, like LMS, IP can be useful in detecting model errors.

Figure 3 shows a two dimensional data set with two outliers where the data points are not uniformly spaced. LS does predictably poorly whereas IP returns the parameters exactly.

Our method extends to nonlinear models as follows: Let D be a differential operator which annihilates the model. Then, proceeding as in the linear case, we replace the test function ϕ by ϕZ. For example, for an exponential model, $y = ae^{\lambda x}$, $ln(y) = ln(a) + \lambda x$. So, $\frac{y'}{y} = \lambda \Rightarrow \frac{(y'')(y)-(y')^2}{y^2} = 0$. So we take,

$$D(i) = D_2(i)y_i - (D_1(i))^2$$

where D_1 is the discrete first derivative, and D_2 is the discrete second derivative. We have also shown [Sturm 1994] how the inner product method can be applied to a nonlinear model for three dimensional eye movements.

Conclusions

We have shown that the integral inner products method is more robust than least squares for simple and multivariate model functions. It compares favorably to more robust methods such as LMS and IRLS for outlying response variables and is computationally simpler. For linear models, no initial guesses for the parameters are required. IP extends naturally to nonlinear models as well.

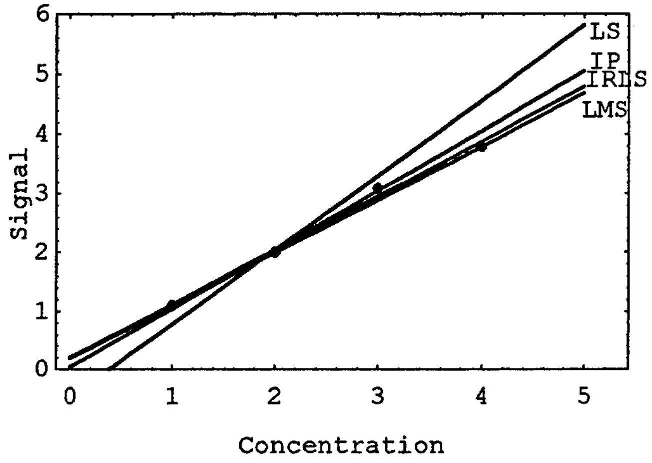


Figure 1. Calibration Data. True relationship: $y = x$.
LS: $1.26x - 0.48$
IP: $1.0x - 0.37$
IRLS: $.92x + .19$
LMS: $.90x + .20$

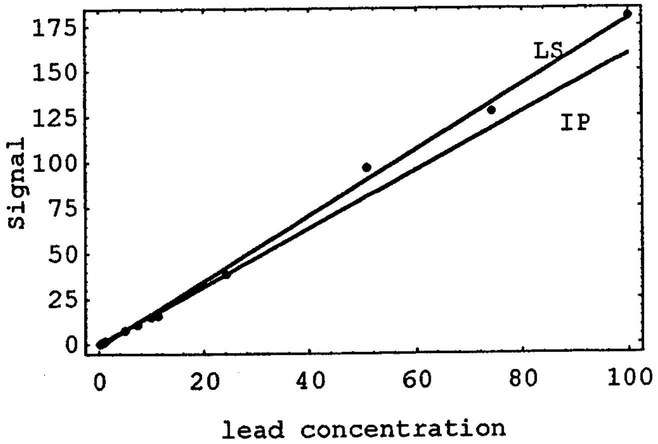


Figure 2. Calibration of lead measurements by plasma emission spectrometry. The difference between LS and IP shows a possible model error.

0	10	0	12	0	0	0	0
8	0	0	11	0	13	0	15
0	8	9	0	11	12	13	0
0	7	0	9	0	11	12	0
5	0	7	(15)	9	0	0	0
0	5	0	7	0	9	(−3)	11
0	4	5	0	0	8	0	10
2	0	0	0	6	0	8	0

Figure 3. Multivariate data. $y = x_1 + x_2$. Outliers are shown in parentheses.
LS: $.76x_1 + 1.08x_2 + .45$
IP: $x_1 + x_2$

References

Barrowdale, I. and Young, A., 1965, *Algorithms for Best L_1 and L_∞ Linear Approximations on a Discrete Set*, Numerische Mathematik, **8**, pp. 295–306.

Bates, D.M., and Watts, D.G., 1988, **Nonlinear Regression Analysis and its Applications**, John Wiley, New York.

Bloomfield, P. and Steiger, W.L., 1983, **Least Absolute Deviations**, Birkhäuser, Boston.

Brownlee, K.A., 1960, **Statistical Theory and Methodology in Science and Engineering**, John Wiley, New York.

Claerbout, J.F., and Muir, F., 1973, *Robust Modeling With Erratic Data*, Geophysics, **38:5**, pp. 826–844.

Clancy, V. J., 1947, , Nature, **159**, pp. 339-340.

Cook, R. Dennis, 1977, *Detection of Influential Observation in Linear Regression*, Technometrics, **19:1**, pp. 15–18.

Daniel, C., and Wood, F.S., 1980, **Fitting Equations to Data**, John Wiley, New York.

Hampel, R.H., Ronchetti, E.M., Rousseeuw, P.J., and Stahel, W.A., 1986, **Robust Statistics**, John Wiley, New York.

Holland, P.W. and Welsch, R.E., 1977, *Robust regression using iteratively reweighted least squares*, Commun. Statist., **A6**, pp. 813–888.

Huber, P.J., 1964, *Robust Estimation of a Location Parameter*, Annals Math. Statist., **35**, pp. 73–101.

Huber, P.J., 1981, **Robust Statistics**, John Wiley, New York.

Marquardt, D.W., 1963, *An Algorithm For Least-square Estimation of Nonlinear Parameters*, SIAM Journal of Applied Mathematics, **11,** pp. 431-441.

Massart, D.L, Kaufman, L., 1986, *Least Median of Squares: A Robust Method for Outlier and Model Error Detection in Regression and Calibration*, Anal. Chimica Acta, **187,** pp. 171–179.

G.R. Phillips and E.M. Eyring, 1983, *Comparison of Conventional and Robust Regression in Analysis of Chemical Data*, Anal. Chemistry, **55,** pp. 1134-1138.

More, J.J., 1977, *The Levenberg-Marquardt Algorithm: Implementation and Theory*, Numerical Analysis: Lecture Notes in Mathematics, **630,** pp. 105-116.

O'Leary, D.P., 1990, *Robust Regression Computation Using Iteratively Reweighted Least Squares*, SIAM J. Matrix Anal. Appl., **11:3,** pp. 466–480.

Phillips, G.R, Eyring, E.M., 1983, *Comparison of Conventional and Robust Regression in Analysis of Chemical Data*, Analytical Chemistry, **55:7,** pp. 1134–1138.

Raphan, T., and Sturm, D., 1991, *Modelling Spatiotemporal Organization of Velocity Storage in the Vestibulo-Ocular Reflex (VOR)*, J. Neurophysiology, **66**, pp. 1410–1421.

Rousseeuw,P.J., and Leroy, A.M., 1987, **Robust Regression and Outlier Detection**, John Wiley, New York.

Sturm, D., *Parameter Estimation for an Eye Movement Model*, International Association for Mathematics and Computers in Simulation (IMACS), World Congress on Computation and Applied Mathematics, July, 1994.

Sturm, D., 1992, *A Noise Resistant Parameter Estimation Method Using Inner Products*, Technical Report: The College of Staten Island, CS9202.

Sturm, D., 1990, *A Nonlinear Identification Method for Modelling the Three Dimensional Structure of Velocity Storage in the Vestibulo-Ocular Reflex (VOR),* Ph.D. thesis, CUNY, June.

variate model functions. It compares favorably to more robust methods such as LMS and IRLS for outlying response variables and is computationally simpler. For linear models, no initial guesses for the parameters are required. IP extends naturally to nonlinear models as well.

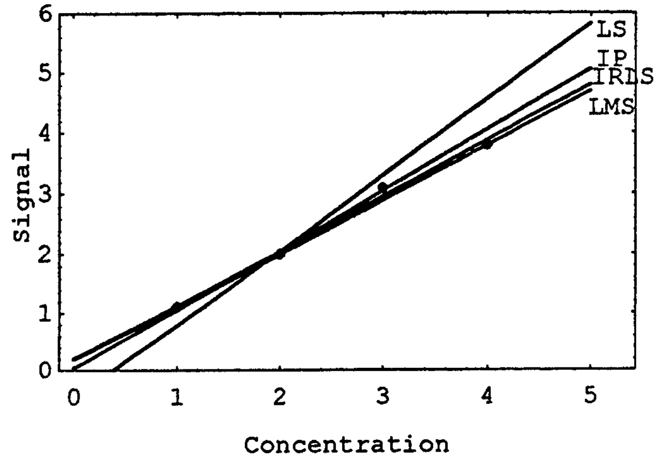


Figure 1. Calibration Data. True relationship: $y = x$.
LS: $1.26x - 0.48$
IP: $1.0x - 0.37$
IRLS: $.92x + .19$
LMS: $.90x + .20$

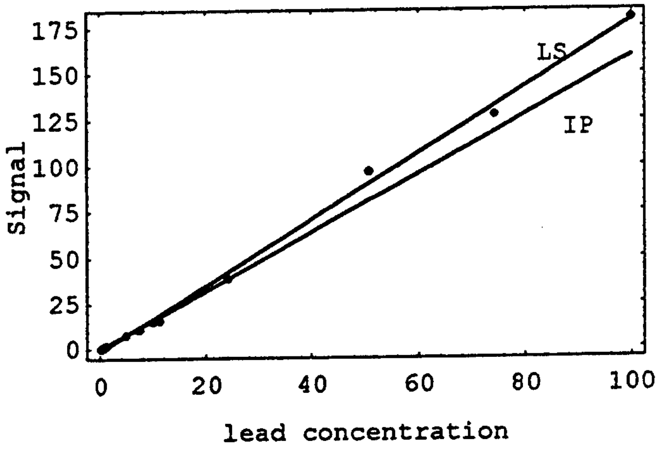


Figure 2. Calibration of lead measurements by plasma emission spectrometry. The difference between LS and IP shows a possible model error.

0	10	0	12	0	0	0	0
8	0	0	11	0	13	0	15
0	8	9	0	11	12	13	0
0	7	0	9	0	11	12	0
5	0	7	(15)	9	0	0	0
0	5	0	7	0	9	(−3)	11
0	4	5	0	0	8	0	10
2	0	0	0	6	0	8	0

Figure 3. Multivariate data. $y = x_1 + x_2$. Outliers are shown in parentheses.
LS: $.76x_1 + 1.08x_2 + .45$
IP: $x_1 + x_2$

References

Barrowdale, I. and Young, A., 1965, *Algorithms for Best L_1 and L_∞ Linear Approximations on a Discrete Set*, Numerische Mathematik, **8**, pp. 295–306.

Bates, D.M., and Watts, D.G., 1988, **Nonlinear Regression Analysis and its Applications**, John Wiley, New York.

Bloomfield, P. and Steiger, W.L., 1983, **Least Absolute Deviations**, Birkhäuser, Boston.

Brownlee, K.A., 1960, **Statistical Theory and Methodology in Science and Engineering**, John Wiley, New York.

Claerbout, J.F., and Muir, F., 1973, *Robust Modeling With Erratic Data*, Geophysics, **38:5**, pp. 826–844.

Clancy, V. J., 1947, , Nature, **159**, pp. 339-340.

Cook, R. Dennis, 1977, *Detection of Influential Observation in Linear Regression*, Technometrics, **19:1**, pp. 15–18.

Daniel, C., and Wood, F.S., 1980, **Fitting Equations to Data**, John Wiley, New York.

Hampel, R.H., Ronchetti, E.M., Rousseeuw, P.J., and Stahel, W.A., 1986, **Robust Statistics**, John Wiley, New York.

Holland, P.W. and Welsch, R.E., 1977, *Robust regression using iteratively reweighted least squares*, Commun. Statist., **A6**, pp. 813–888.

Huber, P.J., 1964, *Robust Estimation of a Location Parameter*, Annals Math. Statist., **35**, pp. 73–101.

Huber, P.J., 1981, **Robust Statistics**, John Wiley, New York.

Jump and sharp cusp detection by wavelets with applications to estimation of functions with jumps

By YAZHEN WANG
Department of Statistics, University of Missouri-Columbia,
Columbia, MO 65211, U.S.A.

Abstract

A wavelet method is proposed to detect jumps and sharp cusps in a function which is observed with noises. We detect jumps and sharp cusps by checking if the wavelet transformation of the data has significantly large absolute values at fine scales. The theory and fast algorithm for the detection are established. For estimating a function with jumps, jump detection is used in construction of a wavelet estimate of the function. The estimate has better visual quality than the direct threshold wavelet estimate (e.g. VisuShrink estimate).

1. Introduction

The recently developed theory of wavelets has drawn much attention from both mathematicians, statisticians and engineers. Orthonormal bases of compactly supported wavelets have been used to estimate functions (see Donoho and Johnstone (1992a, b)). The theory of wavelets permits decomposition of functions into localized oscillating components. This provides an ideal tool to study localized changes such as jumps and sharp cusps in one dimension as well as several dimensions. This paper describes only results about one dimension case in Wang (1994a).

The detection techniques are applied to estimation of a function with jumps. In the seminal work of Donoho (1993a, b), Donoho and Johnstone (1992a, b, c) and Donoho, Johnstone, Kerkyacharian and Picard (1993), orthonormal bases of compactly supported wavelets are introduced to estimate a function. The estimate is the reconstruction of the thresholded empirical wavelet coefficients of the data. This simple estimate enjoys a wide variety of spatial adaptivity and theoretical optimality. If the function has jumps, however, the estimate will have an annoying visual appearance – the reconstruction exhibits many undesirable spurious oscillations near jump locations. Donoho (1993b) considered segmented multiresolution analysis to remedy this drawback. The phenomenon also happens in digital image compression. Because digital image has sharp variations along its edge curves, compression by directly thresholding wavelet coefficients results in oscillations near the edge curves. These oscillations produce so called Gibbs errors at the locations of the edge curves, which degrade considerably the image quality (see Froment and Mallat (1992)). The reason for the phenomenon is explained as follows. When the underlying function has sudden changes such as jumps, the wavelet coefficients are reflected by "large" wavelet coefficients at fine scales near the change locations. After thresholding, only these "large" wavelet coefficients remain at fine levels, and then artificial oscillations which resemble the mother wavelet appear in the reconstruction. The approach here is to divide support of the function into several blocks according to the detected jumps and then use boundary corrected wavelets to estimate the function on each of the blocks. The estimate has a visual advantage that there are no annoying oscillations near the jumps.

The rest of this paper is organized as follows. Sections 2 and 3 introduce the white noise model and wavelet transformation, respectively. Testing hypotheses and estimation are considered in Sections 4 and 5, respectively. Section 6 discusses implementation of the method in practice and Section 7 features an application of jump detection to estimation of functions with jumps. Section 7 presents discussion.

2. The white noise model

Suppose f is observed from the white noise model

$$Y(dx) = f(x)dx + \tau W(dx), \quad x \in [0,1], \qquad (1)$$

where W is a standard Wiener process, and τ is a formal noise level parameter which we think of as small, f is an unknown function which may have jumps and sharp cusps. The problem is to detect these jumps and cusps.

We say a function f has an α-cusp $(0 \leq \alpha < 1)$ at x_0 if there exists a positive constant K such that as $h \downarrow 0$ or $h \uparrow 0$,

$$|f(x_0 + h) - f(x_0)| \geq K|h|^\alpha. \qquad (2)$$

For the case $\alpha = 0$, f has a jump at x_0.

The white noise model (1) is closely related to the following nonparametric regression model:

$$y_i = f(x_i) + \sigma z_i, \quad i = 1, \ldots, n, \qquad (3)$$

with $x_i = i/n$, z_i a standard normal error and $\sigma > 0$ parameter, f an unknown function. Define the regression process $\{Y_n(x) : x \in [0,1]\}$ via $x_0 = 0$, $Y_n(0) = 0$ and $Y_n(x_i) = y_1 + \ldots + y_i$, $i = 1, \ldots, n$, with interpolation between the x_i by Wiener process W for $x_i \leq x < x_{i+1}$. Then Y_n is a white noise process with the function $f_n(x) = f(x_i)$ for $x_i \leq x < x_{i+1}$ and $\tau = \sigma n^{-1/2}$ (see Donoho and Johnstone (1992a)).

3. Wavelet transformation

Let ψ be Daubechies "mother wavelet" (see Chui (1992), Daubechies (1992) and Donoho and Johnstone (1992a,b)), and define $\psi_s(x) = s^{-1/2}\,\psi(x/s)$. The wavelet transformation of f is defined as $Tf(s,x) = \int \psi_s(x-u)f(u)du$. The wavelet transformation $Tf(s,x)$ is a function of the scale (frequency) s and the spatial position (time) x. The plane defined by the pair of variables (s,x) is called the scale-space (or time-frequency) plane.

For compactly supported wavelets, the value of $Tf(s,x)$ depends upon the value of f in a neighborhood of x of size proportional to the scale s. At small scales, $Tf(s,x)$ provides localized information such as local regularity on $f(x)$. For example, if f is differentiable at x, $Tf(s,x)$ has the order $s^{3/2}$, and if f has an α-cusp at x, the maximum of $Tf(s,x)$ over a neighborhood of x of size proportional to the scale s converges to zero at a rate no fast than $s^{\alpha+1/2}$ as s tends to zero (see Daubechies (1992)).

The wavelet transformation of the white noise $W(dx)$ is define to be $TW(s,x) = \int \psi_s(x-u)W(du)$. The wavelet transformation of Y is

$$TY(s,x) = \int \psi_s(x-u)Y(du) = Tf(s,x) + \tau TW(s,x). \qquad (4)$$

At a given scale s, $TW(s,x)$ is a stationary Gaussian process with zero mean and covariance function

$$E(TW(s,x)TW(s,y)) = \int \psi_s(x-u)\psi_s(y-u)du, \qquad (5)$$

and

$$var(TW(s,x)) = \int [\psi(u)]^2 du = 1.$$

Note that $TW(s,x)$ follows a standard normal distribution and that the orders of $Tf(s,x)$ are, respectively, $s^{\alpha+1/2}$ and $s^{3/2}$ for the two cases that $f(x)$ has an α-cusp at x and $f(x)$ is differentiable at x. By (4) we can see that, at a very fine scale s, $TY(s,x)$ is dominated by $\tau TW(s,x)$, while at a coarse scale s, $Tf(s,x)$ dominates $TY(s,x)$. Since the localized information of $f(x)$ is provided by $Tf(s,x)$ at fine scale, if the scale s is too large, the wavelet transformation can not detect local changes with enough precision. Our idea is to select fine scales s_τ such that at those x where $f(x)$ is differentiable, the orders of $Tf(s_\tau, x)$ and $\tau TW(s_\tau, x)$ are balanced. If f has sharp cusps, for x near the locations of the sharp cusps, $TY(s,x)$ will be dominated by $Tf(s,x)$ for $s \geq s_\tau$ and hence significantly larger than the others. Therefore, the sharp cusps will be detected by the wavelet transformation $TY(s,x)$ at the scale levels $s \geq s_\tau$.

Throughout this paper, take η to be any constant that great than 1, and let s_τ be a constant with the exact order $(\tau^2 \, |log\, \tau|^\eta)^{1/(2\alpha+1)}$. Denote by $supp(\psi)$ the support of ψ.

4. Testing hypotheses

Consider the testing problem H_0: f is differentiable against H_1: f has α_i-cusps, $i = 1, \ldots, q$, $q \geq 1$, where $\alpha_i \leq \alpha < 1$.

Our test statistic is the maximum of $TY(s_\tau, x)$ over $0 \leq x \leq 1$. Given a type I error γ, under H_0 the critical value $C_{\tau,\gamma}$ satisfies

$$pr\{\max_{0 \leq x \leq 1} |TY(s_\tau, x)| \geq C_{\tau,\gamma}\} = \gamma.$$

Theorem 1 *If* $0 < \gamma < 1$, *then under* H_0,

$$\lim_{\tau \to 0} pr\{\max_{0 \leq x \leq 1} |TY(s_\tau, x)| \geq C_{\tau,\gamma}\} = \gamma.$$

where

$$\begin{aligned} C_{\tau,\gamma} &= \tau\,(2|log\, s_\tau|)^{-1/2}\,[2|log\, s_\tau| \\ &\quad + log\{(\int [\psi'(u)]^2\, du)^{1/2}/(2\,\pi)\} \\ &\quad - log(-log(1-\gamma)/2)]. \qquad (6) \end{aligned}$$

The proof of Theorem 1 is given in Wang (1994a). Theorem 1 provides a test to check if the underlying function is smooth or has jumps or sharp cusps.

5. Estimation

Because of space we discuss only the case that f has one jump or sharp cusp. See Wang (1994) for multiple jump and sharp cusp detection with known and unknown number of jumps and sharp cusps.

Suppose f has an α-cusp at θ and is differentiable elsewhere. An estimate of θ is the location of the maximum of $|TY(s_\tau, x)|$ over $0 \leq x \leq 1$, that is

$$\hat{\theta} = Arg \max_{0 \leq x \leq 1} \{|TY(s_\tau, x)|\}. \qquad (7)$$

Theorem 2

$$\lim_{\tau \to 0} pr\{\, s_\tau^{-1}\,(\hat{\theta} - \theta) \in supp(\psi)\,\} = 1.$$

The compact support of ψ *implies that the estimate* $\hat{\theta}$ *has the convergence rate* s_τ.

Moreover, suppose that $f(x) = f(\theta) + A_1|x-\theta|^\alpha + o(|x-\theta|^\alpha)$ *as* $x \to \theta$ *if* $\alpha > 0$, *and* $f(\theta+) - f(\theta-) = A_2$ *if* $\alpha = 0$, *where* $A_i \neq 0$, $i = 1, 2$. *Then, as* $\tau \to 0$, $(\hat{\theta} - \theta)/s_\tau$ *converges in probability to the location of the maximum of* $\{|\int \psi(u-t)\,|u|^\alpha\, du\,| : t \in supp(\psi)\}$ *if* $\alpha > 0$, *and* $|\int \psi(u-t)\, sign(u)\, du\,| : t \in supp(\psi)\}$ *if* $\alpha = 0$.

The proof of Theorem 2 is given in Wang (1994a). Theorem 2 establishes asymptotics for the detection. Since jump detection has been studied in the nonparametric regression setting, we compare the convergence rate with those in the literature. By Theorem 2, for the white noise model and the jump point case, $\alpha = 0$, the convergence rate is $\tau^2\, |log\, \tau|^\eta$. Using the relation between the models (1) and (3) described in Section 2 and letting $\tau = \sigma/\sqrt{n}$, we obtain this rate corresponds to the rate $n^{-1}\,(log\, n)^\eta$ for the nonparametric regression model, which are known to be the best possible convergence rates (see Müller (1992)). So the convergence rate s_τ is the best possible rate and the wavelet method is theoretically optimal.

6. Implementation in practice

In practice, we may not have a realization of $Y(x)$ at all points x but discrete observations $Y(i/n)$, $i = 1, \ldots, n = 2^J$. Or equivalently, we observe f from the model (3), that is, $y_i = f(i/n) + \sigma z_i$, $\sigma > 0$, $z_i \sim N(0, 1)$, $i = 1, \ldots, n = 2^J$. The data $y_1, \cdots, y_n$ are discrete and consequently a discrete version of the wavelet transformation must be performed.

The discrete wavelet transformation (DWT) can be written as a n-by-n orthogonal matrix $\mathcal{W}$ which depends on parameters M (number of vanishing moments), S (support width), j_0 (Low-resolution cutoff), and boundary adjustments (see Cohen, Daubechies, Jawerth, and Vial (1993) and Daubechies (1994)). The rows of $\mathcal{W}$ correspond to discretized version of the wavelets ψ_λ. Denote by $W_{jk}(i)$ the i^{th} element of the $(j,k)^{th}$ row of $\mathcal{W}$. Then $\sqrt{n} W_{jk}(i) \approx 2^{j/2}\psi(2^j t)$ for $t = i/n - k\,2^{-j}$ (see Donoho and Johnstone (1992b)).

Let $y = (y_1, \cdots, y_n)$. The DWT of the data y is given by $w = \mathcal{W}y$. Because $\mathcal{W}$ is orthogonal, the inverse DWT is easy and y is recovered from w, that is, $y = \mathcal{W}^T w$. Mallat's pyramidal algorithm (see Chui (1992) and Daubechies (1992)) requires only $O(n)$ operations for computing DWT and reconstruction of DWT.

Using Mallat's pyramid algorithm, we can compute the discrete wavelet transformation of the data (y_i). The elements of w are indexed dyadically as follows: $w_{-1,0}$ and $w_{j,k}, k = 0, \ldots, 2^j - 1, j = 0, \ldots, J-1$. $w_{j,k}$ are corresponding to DWT at the scale levels 2^{-j} and spatial positions $k2^{-j}$, $k = 0, \ldots, 2^j - 1$, $j = 0, \ldots, J-1$. These discrete wavelet transformations are also called empirical wavelet coefficients.

In practice, for a given data set, we can use Mallat's pyramid algorithm to compute the wavelet coefficients and then carry out the detection (see Wang (1994a) for simulations and real examples).

7. An application to estimation of a function with jumps

Suppose we are given n noisy data of f from the regression model (3). The unknown function f has jumps and we want to recover f. The direct threshold estimate (e.g. VisuShrink estimate) will have many undesirable spurious oscillations near jump locations (see Donoho and Johnstone (1992b,c)). The approach here is to divide $[0, 1]$ into several blocks according to the detected jumps and then use boundary corrected wavelets to estimate the function on each of the blocks.

Suppose f has q jumps at $\theta_\ell, \ell = 1, \ldots, q$, where q is a finite integer. For simplicity, suppose q is known (see Wang (1994a) for the case that q is unknown). Let $\hat{\theta}_1, \cdots, \hat{\theta}_q$ be the estimated locations of the jumps and let $I_\ell = [\hat{\theta}_\ell + K\, log\, n/n, \hat{\theta}_{\ell+1} - K\, log\, n/n]$, $\ell = 1, \cdots, q$.

The data are divided into q blocks: $\mathbf{y}_\ell = \{y_i : x_i \in I_\ell\}$, $\ell = 1, \cdots, q$. On each of the I_ℓ, $\ell = 1, \cdots, q$, using boundary corrected wavelets, we compute the wavelet coefficients of $\mathbf{y}_\ell$. Because of the partition, no "large" wavelet coefficients of $\mathbf{y}_\ell$ will appear at fine levels. On each of the I_ℓ, we apply the VisuShrink method (see Donoho and Johnstone (1992b)) to the data $\mathbf{y}_\ell$ to obtain an estimate $\{\tilde{f}_\ell(x_i) : x_i \in I_\ell\}$ of the function $\{f(x_i) : x_i \in I_\ell\}$. The estimate $\tilde{f}_n$ of f on $[0, 1]$ is obtained by pasting $\{\tilde{f}_\ell(x_i) : x_i \in I_\ell\}$ together. The visual advantage of the approach is that $\tilde{f}_n$ has no annoying oscillations near the jumps (see Wang (1994a) for simulations and real examples).

8. Discussion

It is very important to point out that the detection by wavelets has the nature of multiresolution. In the detection we check empirical wavelet coefficients across resolution levels and locate jumps and sharp cusps by empirical wavelet coefficients at these levels. Like smoothing methods with variable smoothing parameters, the multiresolution approach has spatial adaptivity. The detection with spatial adaptivity has many advantages over tradition methods such as kernel methods with fixed bandwidth. For example, a jump is easier to detect than a sharp cusp; because of multiresolution, for fixed sample size and fixed signal - to noise ratio, the jump can be located more accurately by wavelet coefficients in higher resolution levels while the cusp is detected by wavelet coefficients in lower resolution levels.

As a summary, detection by wavelets enjoys theoretical optimality and has fast computational algorithms and can be easily implemented in practice.

Acknowledgements

I would like to express my thanks to David Donoho for suggestion and comment. The research is supported by the Research Board Grant at the University of Missouri and the Summer Fellowship and the Research Council Grant at the University of Missouri - Columbia.

References

Bickel, P. J. and Rosenblatt, M. (1973). On some global measures of deviations of density function estimates. Ann. Statist. **6** 1071-1095.

Chui, C. K. (1992). An Introduction to Wavelets. Academic Press, Boston.

Cline, D. B. H. and Hart, J. D. (1991). Kernel estimation of densities with discontinuities or discontinuous derivatives. Math. Operationsforsch. Statist. Ser. Statist. **22**, 69-84.

COHEN, A., DAUBECHIES, I., JAWERTH, B. and VIAL, P. (1993). Multiresolution analysis, wavelets, and fast algorithms on an interval. Comptes Rendus Acad. Sci. Paris (A). **316** 417-421.

DAUBECHIES, I. (1992). Ten Lecture on Wavelets. CBMS-NSF Series in App. Math., SIAM.

DAUBECHIES, I. (1994). Two recent results on wavelets: Wavelets bases for the interval, and Biorthogonal wavelets diagonalizing the derivative operator. In Recent Advances in Wavelet Analysis, Larry L. Schumaker and Glenn Webb (Ed.). Academic Press, INC. pp. 237-257.

DONOHO, D. L. (1993a). Nonlinear solution of linear inverse problems by wavelet - vaguelette decomposition. Technical Report, Department of Statistics, Stanford University.

DONOHO, D. L. (1993b). Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data. In Different Perspectives on Wavelet, ed. I. Daubechies, American Mathematical Society, Providence, Rhode Island, pp. 173-205.

DONOHO, D. L. and JOHNSTONE, I. M. (1992a). Minimax estimation via Wavelets shrinkage. To appear in Ann. Statist.

DONOHO, D. L. and JOHNSTONE, I. M. (1992b). Ideal spatial adaptation by wavelets shrinkage. To appear in Biometrika.

DONOHO, D. L. and JOHNSTONE, I. M. (1992c). Adapting to unknown smoothness via wavelets shrinking. To appear in J. Amer. Statis. Assoc.

DONOHO, D. L. and JOHNSTONE, I. M., KERKYACHARIAN, G. and PICARD, D. (1993a). Wavelet shrinkage: Asymptopia ? To appear in J. Roy. Statis. Soc. B.

ENGLE, R. F., GRANGER, G. W. J., RICE, J. and WEISS, A. (1986). Semiparametric estimates of the relation between weather and electricity sales. J. Amer. Statist. Assoc. **81**, 310-320.

EUBANK, R. L. and SPECKMAN, P. L. (1992). Nonparametric estimation of functions with jump discontinuities. To appear in Proceedings of the Mount Holyoke Conference.

FROMENT, J. and MALLAT. S. (1992). Second generation compact image coding with wavelets. In Wavelet and their application, ed. Ruskai et al., Jones and Bartlett, Cambridge, MA, pp. 655-677.

GROSSMANN, A. (1986). Wavelet transformation and edge detection. In Stochastic processes in Physics and engineering, M. Hazewinkel, editor. Dodrecht: Reidel. pp.149-157.

MALLAT, S. and HWANG, W. L. (1992). Singularity detection and processing with wavelets IEEE Trans. on Information Theory **38**, 617-643.

MÜLLER, H. G. (1992). Change-points in nonparametric regression analysis. Ann. Statist. **20**, 737-761.

MÜLLER, H. G. and SONG, K. (1992a). On the estimation of multi-dimensional boundaries. Technical Report, Division of Statistics, UC-Davis.

MÜLLER, H. G. and SONG, K. (1992b). On multidimensional change-points in regression. Technical Report, Division of Statistics, UC-Davis.

MÜLLER, H. G. and SONG, K. (1992c). Asymptotics for change-point estimators in smooth regression models. Technical Report, Division of Statistics, UC-Davis.

MÜLLER, H. G. and SONG, K. (1993). Cube splitting in multidimensional edge estimation. Technical Report, Division of Statistics, UC-Davis.

WAHBA, G. (1986). Partial spline models for the semi-parametric estimation of functions of several variables. In Statistical Analysis of Time Series, 319-329, Tokyo: Institute of Statistical Mathematics.

WANG, Y. (1994a). Jump and sharp detection by wavelets. Manuscript.

WANG, Y. (1994b). Function estimation via wavelets for data with long - range dependence. Manuscript.

YIN, Y. Q. (1988). Detection of the number, locations and magnitudes of jumps. Comm. Statist. Stochastic Models **4**, 445- 455 .

NUMERICAL TECHNIQUES IN DISTRIBUTION FITTING

Vic Hasselblad

Center for Health Policy Research and Education

Duke University, Durham, NC 27708

1 Abstract

Fitting distributions to data has many applications. First, most analyses make assumptions about the distribution of the data or the residuals, and these assumptions should be checked whenever possible. Second, inferences about the fraction of observations above or below some particular point are often needed, and modeling the distribution may be the best way to obtain these estimates. Third, biological models often require the parameter estimates from a distribution. There are several numerical methods used in distribution fitting, and this paper will discuss three: the standard Newton-Raphson method, a modified Gauss-Newton method, and the EM Algorithm.

2 Introduction

This paper restricts its discussion to maximum likelihood techniques. Distributions with range parameters to be estimated are not considered because of the difficulties in estimating these parameters using maximum likelihood methods. Several different numerical methods are commonly used to derive estimates, and this paper will discuss three of these: the standard Newton-Raphson method, a modified Gauss-Newton method, and the EM Algorithm. The performance of these methods for problems of distribution fitting will be discussed.

Even with these restrictions, there are several problems in distribution fitting which are challenging. These include grouping and censoring of data, truncation of distributions, and mixtures of distributions. Truncation differs from censoring in that the values below (or above) some cutoff point are never seen. For example, lengths of fish gathered by net will not include fish below a certain size since those fish pass through the net. The number of fish below this size is never known. With censored data, information is available on the number below the limit. Censoring is a common problem with environmental pollutants which often exist in concentrations below the minimum detectable limit of the measuring instrument (left censoring). Censoring also occurs in time-to-tumor studies where some animals die before they get a tumor (right censoring). Grouping occurs when the number of observations is so large that it is impractical to retain the individual values.

3 Newton-Raphson Method

The Newton-Raphson method is described in every book on numerical analysis (e.g., see Burden and Faires, 1985). Unfortunately, it performs rather poorly for many distributional fitting problems. However, one particular problem for which it is very useful is in estimating the inverse of a cumulative distribution function, F. Assume that we want to solve for x given the probability p and the parameters α and β:

$$x = F^{-1}(p,\alpha,\beta)$$

$$\text{or equivalently} \quad F(x,\alpha,\beta) - p = 0.$$

The Newton-Raphson iteration scheme becomes

$$x^{(k+1)} = x^{(k)} - \frac{F(x^{(k)},\alpha,\beta) - p}{f(x^{(k)},\alpha,\beta)}$$

where $x^{(k)}$ denotes the k$^{\text{th}}$ estimate of x and f is the probability density function.

As an example, consider the problem of evaluating the inverse incomplete beta function. The incomplete beta function is given by

$$p = F(x) = \int_0^x \frac{t^{\alpha-1}(1-t)^{\beta-1}\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\,dt,$$

where $\Gamma(\alpha)$ is the gamma function. The problem is to solve for x when p is specified. Assume $p = 0.025$, $\alpha = 3.5$, and $\beta = 12.5$. Using the approximate formula in Abramowitz and Stegun (p. 945) for an initial estimate, the following are the estimates by iteration.

Iteration	x	$F(x,\alpha,\beta) - p$
0	0.06643	0.0090519
1	0.06033	0.0006947
2	0.05978	0.0000056
3	0.05977	0.0000000

This is the standard quadratic convergence expected from the Newton-Raphson method.

4 Modified Gauss-Newton Method

A very good general method for obtaining a maximum likelihood estimates of the parameters of a distribution is a modified Gauss-Newton method. This method is normally applied to least squares estimation, but it has some advantages for maximum likelihood estimation. There appear to be no references which describe the procedure given below exactly -- the closest is that of Berndt et al. (1974). The following notation will be used in the estimation formulas. Let the ith observation be denoted by x_i. In general, the log-likelihood function for continuous data can be written as

$$L = \sum_{i=1}^{n} Log(f_i),$$

where f_i is the density function for the distribution evaluated at x_i, $f(x_i)$. If a single distribution is truncated, then f_i is given by

$$f_i = f(x_i) / [F(B) - F(A)], \text{ where}$$

A is the left truncation point, B is the right truncation point, and $f(x_i)$ is the complete (not truncated) distribution evaluated at x_i, and F(x) is the cumulative distribution function. The first partial derivative can be written as

$$\frac{\partial \log(f_i)}{\partial\theta} = \frac{\frac{\partial f(x_i)}{\partial\theta}}{f(x_i)} - \frac{\frac{\partial F(B)}{\partial\theta} - \frac{\partial F(A)}{\partial\theta}}{F(B) - F(A)}.$$

The method involves estimating the expected values of the second partial derivatives using first partial derivatives. Using the results of Cramér (1946, p. 502), we know that

$$E[\nabla L]^2 = -E[\nabla(\nabla L)'].$$

where ∇ is the gradient function. This suggests replacing the matrix of second partial derivatives with the expected value of the first partial derivatives squared. The calculation of the expected value is difficult, but we can finesse this problem by using the observed sample distribution instead of taking expectations. Asymptotically, the distribution of our sample will approach the true (but unknown) distribution. Furthermore, the sum of cross products of the first partial derivatives will be guaranteed to be positive definite. The estimated matrix is inverted and used in the standard Newton formulation. For some problems, the change is estimates may be too large, resulting in a decrease rather than an increase in the likelihood. In those cases, it is necessary to successively chop the change in half until the likelihood is increased. Iteration is stopped when the change in the parameter values is arbitrarily small.

To demonstrate the method, consider the truncated logistic distribution (also known as the sech-squared distribution). One form of the density is given by

$$f(x) = e^{-(x-\alpha)/\beta}/\beta[1 + e^{-(x-\alpha)/\beta}]^2 \text{ where } \beta > 0.$$

It is often derived from a differential equation. The cumulative logistic distribution is defined as

$$F(x) = 1/[1 + e^{-(x-\alpha)/\beta}].$$

It is commonly used in dose-response analysis, but the use of the logistic distribution for distribution fitting is less common. The distribution is similar in shape to the normal distribution, but has more mass in the tails. Thus it may be a good alternative to the normal distribution in situations where the data are symmetric, but the tails are heavier.

In order to use the modified Gauss-Newton method, the required partial derivatives are:

$$\left.\frac{\partial f(x)}{\partial\alpha}\right|_x = \frac{1}{\beta}\left(\frac{1 - e^{-(x-\alpha)/\beta}}{1 + e^{-(x-\alpha)/\beta}}\right)f(x),$$

$$\left.\frac{\partial f(x)}{\partial\beta}\right|_x = -\frac{1}{\beta}\left(1 - \frac{1 - e^{-(x-\alpha)/\beta}}{\beta(1 + e^{-(x-\alpha)/\beta})}\right)f(x),$$

$$\left.\frac{\partial F}{\partial\alpha}\right|_x = -f(x) \text{ and}$$

$$\left.\frac{\partial F}{\partial\beta}\right|_x = \frac{-(x - \alpha)f(x)}{\beta}.$$

The asymptotic covariance matrix can also be calculated from the same inverse matrix of partial derivatives.

To illustrate the method, consider the data of Kenyon, Scheffer, and Chapman (1954) on the Pribilof fur-seal herd in Alaska. The herd has been on the verge of extermination several times. During the commercial sealing season, male seals whose length (tip of snout to base of tail) is between 41 and 45 inches (to the nearest inch) are killed (clubbed to death). Note that the data are naturally truncated at 40.5 and 45.5 inches (see Figure 1). The estimates of alpha, beta, and the log-likelihood by iteration are in the following table.

Iteration	α	β	log-likelihood
0	42.328	0.8398	-651.3232
1	41.960	0.9558	-587.3418
2	41.835	0.9580	-586.6463
3	41.846	0.9483	-586.6409
4	41.843	0.9497	-586.6407
5	41.844	0.9494	-586.6407

Note the rapid convergence, comparable to standard quadratic methods. The actual fit is shown Figure 1.

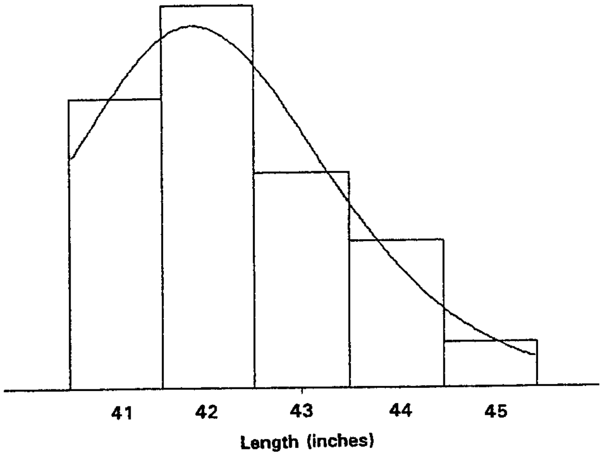


Figure 1. Fit of a truncated logistic distribution to lengths of fur seals in the Pribilof Islands.

5 EM Algorithm

Another method for estimating the parameters of a distribution is the EM algorithm as described by Dempster, Laird, and Rubin (1977). This method works well for the problems created by mixtures and grouping, and can be described as follows

E step: estimate the expected values of the sufficient statistics for the missing data using the current estimates of the parameters.

M step: recompute the estimates the parameters from the expected values of the sufficient statistics. Iterate by returning to the E step until the desired accuracy is attained.

Grouped data are common and yet standard Shephard correction estimates can be quite biased. As an example, consider the grouped blood lead data shown in Figure 2. The city of New York conducted a blood lead screening program in children for several years to prevent lead poisoning from paint chips (Billick et al., 1970). A detailed discussion of the analysis of this data set was given by Hasselblad et al., 1980. Because the data were collected for screening purposes, the actual blood lead values (in micrograms per deciliter) were not recorded, but were grouped into ten unit intervals (except for the first interval). The data from the years 1970 to 1973 showed both larger means and larger standard deviations than did data from later years. The larger means were undoubtedly due to the lead in air and dust resulting from the high lead content in gasoline. The larger standard deviations were probably due to the seasonal changes in automobile travel and outdoor activities of the children. After most lead was removed from gasoline, the means of both the blood lead and air lead levels dropped. Figure 2 shows the data for 1974.

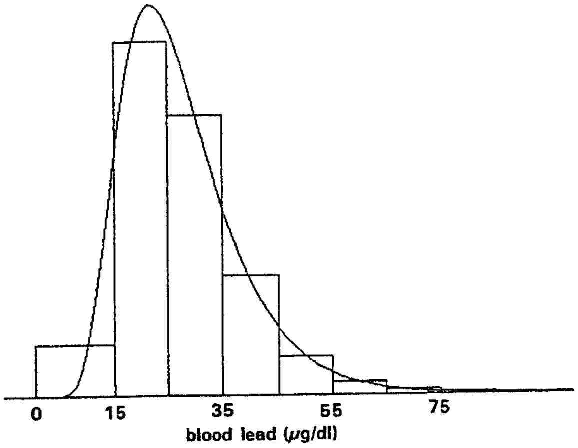


Figure 2. Fit of a grouped lognormal distribution to blood lead levels in black children aged 1-3 in New York City in 1974.

In order to use the EM algorithm, the expected value of x and x^2 assuming a normal (after transformation) distribution must be calculated for each interval. If the endpoints of the intervals are denoted as c_i, then these expectations are given by

$$E(x \mid c_{i-1}<x<c_i) = \mu - \sigma^2 z_{1i}$$

$$E(x_i^2 \mid c_{i-1}<x<c_i) = \sigma^2(1 + z_{2i}) + \mu^2 - \mu z_{1i}\sigma^2$$

where

$$z_{1i} = f(c_i) / F(c_i) - f(c_{i-1}) / F(c_{i-1}) \quad \text{and}$$

$$z_{2i} = c_i f(c_i) / F(c_i) - c_{i-1} f(c_{i-1}) / F(c_{i-1}).$$

This same technique can be used to fit a linear model to grouped data (see Hasselblad et al., 1980). For grouped data problems, the EM algorithm converges quite quickly, and the estimates and log-likelihood by iteration are shown in the following table.

Iteration	μ	σ	log-likelihood
0	3.1629	0.4823	-3337.0311
1	3.2033	0.4118	-3265.2098
2	3.2121	0.3968	-3260.8766
3	3.2143	0.3934	-3260.6353
4	3.2148	0.3926	-3260.6220
5	3.2149	0.3924	-3260.6212
6	3.2150	0.3924	-3260.6212

A more difficult problem is the estimation of parameters from mixtures of distributions. The algorithm is quite simple and was given by Hasselblad (1966). Part of the numerical problem results from the likelihood function itself. For example, Ali and Giaccotto (1982) show data for the logarithm of the ratio of consecutive monthly prices of Bethlehem Steel stock over a ten year period (shown in Figure 3).

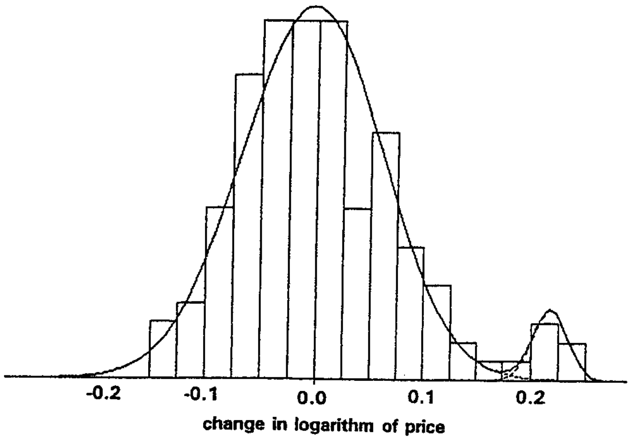


Figure 3. Fit of a mixture of two normal distributions to the logarithm of the ratio of consecutive monthly prices of Bethlehem Steel stock.

Because the data are already log-transformed and show two peaks, it is logical to fit a mixture of two normal distributions to the data. If we look at a graph of the log-likelihood function as a function of μ_2 and σ_2 with p, μ_1, and σ_2 fixed at their maximum likelihood estimates, we get the graph shown in Figure 4.

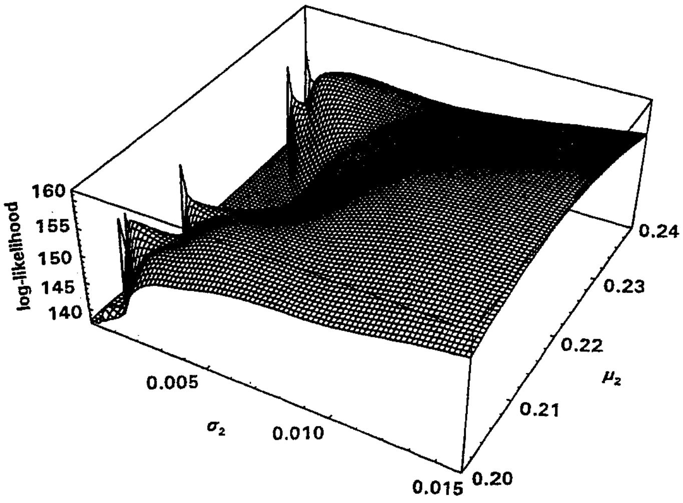


Figure 4. Graph of the log-likelihood function as a function of μ_2 and σ_2 with p, μ_1, and σ_2 fixed at their maximum likelihood estimates for monthly prices of Bethlehem Steel stock.

The spikes on the left actually go to infinity where to μ_2 takes on the value of any data point and $\sigma_2 \to 0$, a fact which was reported by Day (1969). Thus we can only hope to find a relative maximum of the likelihood function. Furthermore, the likelihood function can be very flat over regions near the solution (see the right hand part of Figure 4 which is close to the maximum likelihood solution).

This is a difficulty for any method, but the EM algorithm will continue to move to a relative maximum of the likelihood without and step size correction even under these difficult conditions. A graph of the likelihood as a function of the number of iterations for the Bethlehem Steel stock data is shown in Figure 5. Note that once the estimates get reasonable close (after 1960 iterations), they zoom in quite quickly to the solution. The fitted curve was shown in Figure 3.

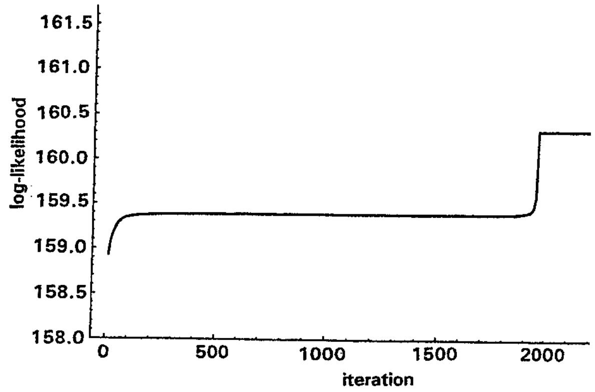


Figure 5. Graph of the log-likelihood function as a function of the number of iterations for monthly prices of Bethlehem Steel stock.

As indicated earlier, the blood lead data of New York City in the years prior to 1974 tended to have extra variation. For the data of 1972, the fit to a mixture of two lognormal distributions is shown in Figure 6. This is an application of the EM algorithm to solve both the problems of a mixture as well as grouping in the same example. Although there is no visual evidence of a mixture, the likelihood ratio test for the improvement in fit over a single lognormal gives a chi-square of 65.914 for three degrees of freedom (p < 0.00001). The existence of the two distributions may be an artifact of the seasonal variation in lead exposure which was not included as a covariate in the analysis.

As might be expected, the convergence to the maximum likelihood estimate is quite slow using the EM algorithm, taking approximately 2500 iterations to converge.

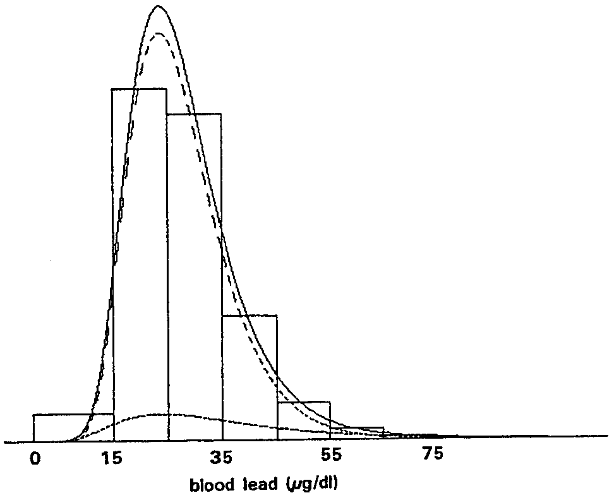


Figure 6. Fit of a grouped lognormal distribution to blood lead levels in black children aged 1-3 in New York City in 1974.

6 Discussion

The three numerical techniques just discussed have proved to be useful in distribution fitting. The modified Gauss-Newton method as described here is useful for a wide range of distributions, including truncated distributions. In conjunction with the EM algorithm, it will handle grouped distributions. For mixtures of distributions, the EM algorithm is superior, although it can converge very slowly.

In combination, these methods can estimate parameters from normal, lognormal, exponential, gamma, Weibull, logistic, extreme value, and beta distributions, as well as mixtures of normal, lognormal, and exponential distributions. For information on software using these techniques for distribution fitting contact the author.

7 References

Abramowitz M. and I. E. Stegun (1972). Handbook of Mathematical Functions, U.S. Government Printing Office, Washington D.C., 1972.

Ali, M. M. and C. Giaccotto (1982). "The Identical Distribution Hypothesis from Stock Market Prices - Location and Scale-Shift Alternatives", Journal of the American Statistical Association, Vol. 77, pp. 19-28.

Berndt, E., B. Hall, R. Hall, J. and Hausman (1974). "Estimation and Inference in Nonlinear Structural Models. Annals of Economic and Social Measurement, Vol. 3, pp. 655-665.

Billick, I.H., A.S. Curran, D.R. Shier (1970). "Analysis of Pediatric Blood Lead Levels in New York for 1970 - 1976", Archives of Environmental Health, Vol. 31, pp. 180-190.

Burden, Richard L. and J. Douglas Faires (1985). Numerical Analysis, Prindle, Weber, & Schmidt, Boston.

Cramér, Harald (1946). Mathematical Methods of Statistics, Princeton University Press, Princeton.

Day, N.E. (1969). "Estimating the Components of a Mixture of Normal Distributions", Biometrika, Vol. 56, pp. 463-474.

Dempster, A. D., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data Via the EM Algorithm", Journal of the Royal Statistical Society B, Vol. 38, pp. 1-22.

Hasselblad, Victor (1966). "Estimation of Parameters for a Mixture of Normal Distributions", Technometrics, Vol. 8, pp. 431-441.

Hasselblad, Victor, Andrew G. Stead, and Warren Galke (1980). "Applications of Regression Analysis to Grouped Blood Lead Data", Journal of the American Statistical Association, Vol. 75, pp. 771-778.

Kenyon, K. W., V. B. Scheffer, and D. G. Chapman (1954). A Population Study of the Alaska Fur-Seal Herd, U. S. Department of the Interior, Washington, D. C.

Picciotto, R. (1970). "Tensile Fatigue Characteristics of a Sized Polyester/Viscose Yarn and Their Effect on Weaving Performance", unpublished Master's thesis, North Carolina State University, Raleigh, NC.

Maximum Likelihood Density Estimation with Term Creation and Annihilation

J. L. Solka, C. E. Priebe, G. W. Rogers, W. L. Poston, and R. A. Lorey
Naval Surface Warfare Center Dahlgren Division
Systems Research and Technology Department
Advanced Computation Technology Group, Code B10
Dahlgren, VA 22448-5000

Abstract

This paper presents some preliminary results on a new method of density estimation that models an unknown distribution as a mixture of normal components. The model is first built using the recursive adaptive mixtures procedure of Priebe. This preliminary model is allowed to come to equilibrium with the data via a sequence of term annihilations based on the Aikaike Information Criterion in combination with adjustment of the remaining model parameters via the standard expectation maximization algorithm.

Introduction

Given $X=\{x_1, x_2, ..., x_n\}$ where each x_i is i.i.d. according to an unknown density f(x) then one is often interested in estimating f(x). This problem occurs in such areas as exploratory data analysis, classification, and regression. There are a variety of approaches to the multivariate density estimation problem[1].

An often used parametric approach is that of finite mixture models[2] in combination with the expectation maximization (EM) method of Dempster, Laird, and Rubin[3]. One difficulty with this tactic is that one needs some idea as to the appropriate number of terms in the mixture model. Given this information the EM algorithm is guaranteed to convergence to at least a local maxima in the likelihood surface.

Some of the previous nonparametric approaches include histograms [4], frequency polygons [5], adaptive histograms[6], average shifted histograms [7], and kernel estimators [8]. These approaches are beneficial in that they possesses nice asymptotic consistency properties and robustness with regard to nonnormality. They are at a disadvantage as compared to the mixture model approach when it is suspected that the unknown true density is a mixture of a number of components and one would like to estimate the posteriori probability of underlying component membership for an unlabeled observation.

This type of problem exists in the areas of medical diagnosis and image processing. In medical diagnosis the component membership may play an important role in identification of the underlying mechanism of disease or the identification of appropriate tissue type in an image. In the general problem of image analysis the component membership may pertain to region type.

A recently developed density estimation technique that circumvents some of the problems of the above techniques is the adaptive mixtures procedure of Priebe and Marchette [9]. This procedure is a blend of the finite mixtures and kernel estimator approach. It is essentially a mixtures type approaches that allows for the creation of new terms as indicated by the data complexity. We have successfully applied this technique in combination with fractal-based features to the detection of man-made objects in land[10] and aerial[11] images, the general problem of texture classification[12], and the measurement of breast parenchymal tissue density[13]. The adaptive mixtures estimator is asymptotically consistent like the kernel estimator, but it has the added benefit of creating additional terms at a rate which is considerably less then the rate n creation associated with the kernel estimator.

One drawback to the adaptive mixtures estimator is that even though there is asymptotic L1 convergence for the procedure there is no finite sample or asymptotic assurance that the match between the complexity of the final model and the data is optimal. Another way of saying this is that if the underlying distribution is a finite mixture of M terms, one would like M terms in the adaptive mixtures solution. The goal of our work is to modify the adaptive mixtures procedure so that it produces a model that not only matches the unknown density in a functional sense, but also in terms of model complexity.

Approach

Given an unknown distribution $\alpha(x)$ we seek to model the distribution using $\alpha^*(x)$ defined by

$$\alpha^* (x;\Psi) = \sum_{i=1}^{m} \pi_i K(x;\Gamma_i) , \qquad (1)$$

where K is some fixed density parameterized by Γ, and $\Psi = (\pi_1, \Gamma_1, \pi_2, \Gamma_2, ..., \pi_m, \Gamma_m)$. The π_i's are referred to as the mixing proportions. (We can assume for much of what follows that K is taken to be the normal distribution, in which case Γ_i becomes $\{\mu_i, \sigma_i\}$.) In the simplest case the mixture is assumed to have a single term and the parameters that need to be estimated are the mean and covariance of the distribution. It is important to note that unlike finite mixture models the number of terms m is not fixed but is driven by the data.

The basic stochastic approximation approach is to recursively update the estimate Ψ^* of the true parameters Ψ_0 based on the latest estimate Ψ_t^* and the newest data point x_{t+1}. That is,

$$\Psi_{t+1}^* = \Psi_t^* + \Phi_t(x_{t+1};\Psi_t^*) \qquad (2)$$

for some update function Φ_t. This approach is usually used when it is known that the true distribution is a finite mixture of components. However, one can certainly approach the problem from the perspective of fitting the data to a finite mixture model where one finds the Ψ_{t+1}^* that produces the best fit.

The specific form of the update equation that we use is the one suggested by Titterington [14]. If we let $I(\Psi)$ be the Fisher information then the version of the recursive update formula we will use is

$$\Psi^*_{t+1} = \Psi^*_t + (nI(\Psi^*_t))^{-1} (\frac{\partial}{\partial\Psi}) \log (\alpha^* (x_{t+1};\Psi^*_t)) \quad (3)$$

where the derivative represents the vector of partial derivative with respect to the components of Ψ.

The AMDE stochastic approximation approach is to recursively update Ψ^*, the estimate of the true parameters Ψ_0, while at the same time providing the capability to expand the extent of the parameter space Ψ^* if dictated by the underlying complexity of the data. We note that in the AMDE case our parameter space Ψ is given by $(\pi_1,\theta_1, \pi_2,\theta_2, ...,\pi_n,\theta_n, ...)$. The procedure

$$\Psi_{t+1}^* = \Psi_t^* + A^*U_t(x_{t+1};\Psi_t^*) + B^*C_t(x_{t+1};\Psi_t^*,t), \quad (4)$$

is used to recursively update the density where $A=[1-P_t(x_{t+1};\Psi_t^*)]$, and $B=P_t(x_{t+1};\Psi_t^*)$. P_t represents a possibly stochastic create decision and takes on values 0 or 1. U_t updates the current parameters using (3) while C_t adds a new component to the model. As is implicit in the equation, the decision to add a new term is a function of the current data point, our current estimation of the parameters, and time. The time dependence is important in those cases that we wish to anneal the probability of creation as a function of training time.

Previous work in the literature has examined the application of the Akaike Information Criterion (AIC) [15] to the determination of the number of components in a finite mixture[16]. The AIC estimates the expected value of the Kullback-Leibler information between the estimated model and the unknown true density

$$AIC = E[KL(\alpha, \alpha^*)] = \int \alpha \log \frac{\alpha}{\alpha^*} . \qquad (5)$$

AIC is defined in terms of likelihood, L, and the number of free parameters in the model, M, as

$$AIC(f) = -2ln(f(\dot{x})) + 2M. \qquad (6)$$

Using this idea as a starting point we have developed a procedure that uses a single or set of adaptive mixtures density estimates and produces a pruned model with a lower complexity. This procedure uses AIC to evaluate the appropriateness of lower complexity models that have been subjected to the iterative EM method. In the iterative EM method the update equation takes the form

$$\Psi_{t+1}^* = \Psi_t^* + \Phi(\underline{x};\Psi_t^*), \qquad (7)$$

where Φ is the update function and $\underline{x}$ is the set of observations.

Our approach to the pruning process is as follows. Given α^*_k an initial adaptive mixtures approximation to α containing k terms the AIC of each of the k-1 term models is computed after application of the EM method of equation 7 to each of the models. If $AIC(\alpha^*_{k-1}) < AIC(\alpha^*_k)$ for one of the k-1 term models then the pruning process is repeated using this model. This process of pruning and expectation maximization is repeated until no further improvement is possible. It is important to point out that at each pruning step the remaining terms π's are updated based on their Mahalonobis distance to the pruned term prior to relaxation with the EM method.

Results

This pruning approach was tested on data sets drawn from two different bimodal two term distributions and from one four mode four term distribution, see Figure 1. In each case 10,000 points were drawn from each distribution. Twenty-five bootstrap resamples were extracted from each of the data sets. A ten term adaptive mixtures model was created for each of the resampled data sets. Each of these models were then subjected to the AIC based pruning process. This process provides a model complexity distribution based on the data set.

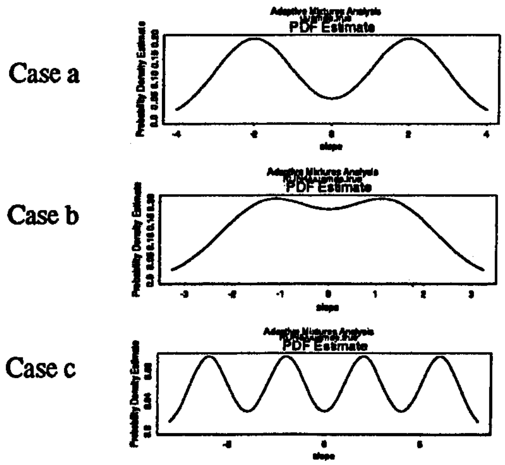
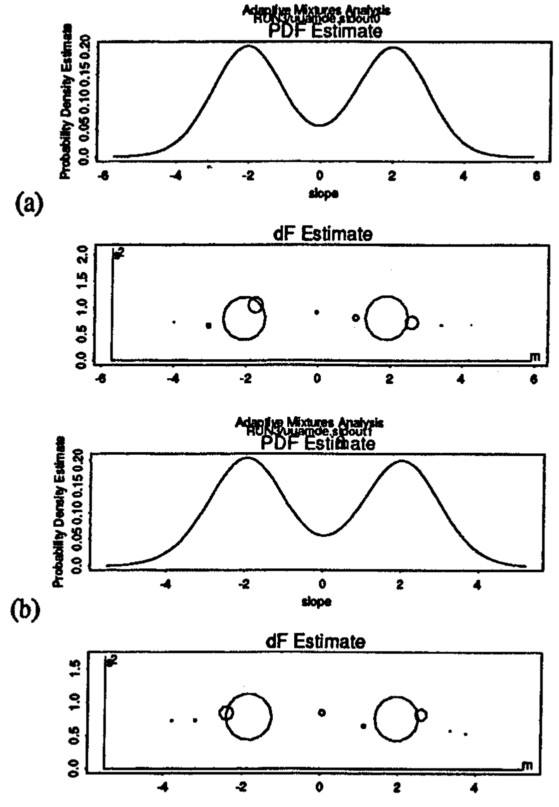


Figure 1 - $\alpha(x)=.5*N(-2,1)+.5*N(2,1)$,
$\alpha(x) =.5*N(-1.25,1)+.5*N(1.25,1)$,
$\alpha(x) =.25*N(-6,1)+.25*N(-2,1)+.25*N(2,1)$
$+.25*N(6,1)$



Figures 2 a and b - Adaptive mixtures estimates for two of the resamplings of the data set drawn from case a.

In Figures 2a and 2b we present adaptive mixtures solutions for two of the resamplings of the data set drawn form $\alpha(x)=.5*N(-2,1)+.5*N(2,1)$. We have included dF space plots along with the standard functional representation of the distributions. dF space plots are an effective way to display the terms in a mixture. Each term $\pi_i N(\mu_i,\sigma_i^2)$ is plotted as a circle whose radius is proportional to π_i and whose center is given by (μ_i,σ_i^2). We notice that the terms in each of the two solutions are markedly different. This phenomena falls under the adage that there is "more then one way to skin a cat." We also notice that there are more then the "theoretical" number of terms needed. Each of the models is made up of ten terms. The occurrence of a matching number of terms in each model is the result of our initial constrainment of the model complexity.

Figure 3 illustrates the results of the pruning process. For each of the three distributional types we have plotted a histogram of the number of terms in the final pruned models for each of the twenty-five resamples. In case a the procedure converged to the correct solution 11 of 25 times. In case b the procedure converged to the correct solution 7 of 25 times, and 17 of 25 times in case c. In all cases models of more appropriate complexity are provided by the procedure.

The last thing left to be discussed is the output of the pruning procedure. In Figures 4 a, b, and c we present an expectation maximized adaptive mixture solution along with the output of pruning this solution. We notice that the number of terms in the solution has been reduced from ten to the appropriate number in each case. We also notice that the terms left from the process are in approximately the correct location and have about the right mixing coefficients and variances.

Summary

We have presented some very preliminary results concerning a new nonparametric density estimation technique that combines the flexibility of term creation with the parsemoneousness of term annihilation. Much work needs to be done to strengthen these initial anecdotal results.

References

1. Scott, D. W. (1992). *Multivariate Density Estimation.* John Wiley and Sons, New York, NY.

2. Everitt, B. S. and Hand, D. J. (1981). *Finite Mixture Distributions.* Chapman and Hall. London, UK.

3. Dempster, A. P., Lairs, N. M., and Rubin, D. B. (1977) "Maximum Likelihood from Incomplete Data via the EM Algorithm" *J. Royal Statist. Soc., Series B* 39 1-38.

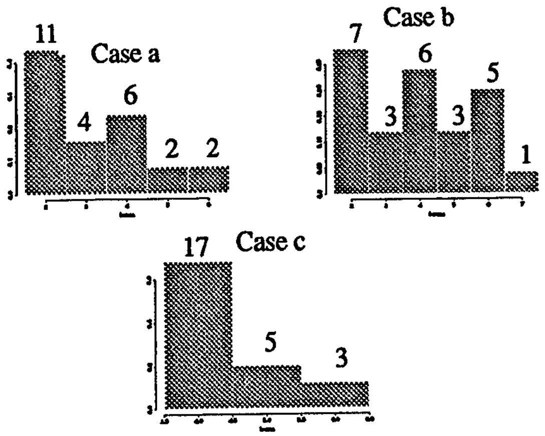
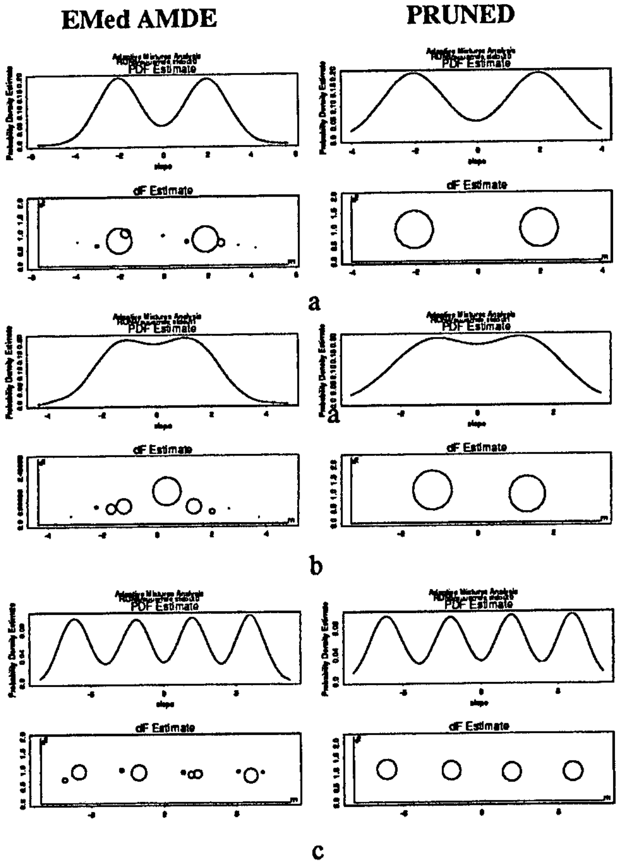


Figure 3 - Histograms of the number of terms in the final pruned models for each of the three test cases.



Figures 4 a,b, and c - Expectation maximized adaptive mixtures estimates along with the output of the pruning process for each of the three distributional cases.

4. Sturges, H. A. (1926). "The Choice of a Class Interval" *J. Amer. Statist. Assoc.* 21 65-66.

5. Scott, D. W. (1985). "Frequency Polygons" *J. Amer. Statist. Assoc.* 80 348-354.

6. Wegman, E. J. (1970). "Maximum Likelihood Estimation of a Unimodal Density Function" *Ann. Statist.* 41 457-471.

7. Scott, D. W. (1985). "Average Shifted Histograms: Effective Nonparametric Density Estimation in Several Dimensions" *Ann. Statist.* 13 1024-1040.

8. Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall. New York, NY.

9. Priebe, C.E. and Marchette, D.J. (1993). "Adaptive Mixture Density Estimation" *Pattern Recognition,* Vol. 26, No. 5, 771-785.

10. Solka, J.L., Priebe, C.E., and Rogers, G.W. (1992) "An Initial Assessment of Discriminant Surface Complexity for Power Law Features" *Simulation* Volume 58, No. 5, pp 311-318.

11. C. E. Priebe, J. L. Solka, and G. W. Rogers (1993) "Discriminant Analysis in Aerial Images Using Fractal Based Features" in *Adaptive and Learning Systems II, F. A. Sadjadi, Ed., Proc. SPIE 1962*, pp. 196-208.

12. J. L. Solka, C. E. Priebe, and G. W. Rogers (1993) "A Probabilistic Approach to Fractal Based Texture Discrimination" in *Adaptive and Learning Systems II, F. A. Sadjadi, Ed., Proc. SPIE 1962*, pp. 209-218.

13. Priebe, C. E., Solka, J. L., Lorey, R. A., Rogers, G. W., Poston, W. L., Kallergi, M., Qian, W., Clarke, L. P., Clark, R. A. (1994) "The Application of Fractal Analysis to Mammographic Tissue Classification" *Cancer Letters*, 77, pp. 183-189.

14. D. M. Titterington (1984) "Recursive parameter estimation using incomplete data", *J. Royal Stat. Soc., Ser. B, Vol. 46*, pp 257-267.

15. Akaike, H. (1974). "A New Look at the Statistical Model Identification" *IEEE Trans. Auto. Control*, vol.19, pp.716-723.

16. Liang, Z., Jaszczak, R. J., and Coleman, R. E. (1992). "Parameter Estimation of Finite Mixtures Using the EM Algorithm and Information Criteria with Application to Medical Image Processing" *IEEE Transactions on Nuclear Science*, Vol. 39, No. 4, pp. 1126-1133.

Parseval Quadrature for Computing Normal Tail Probablitities

George R. Terrell
Statistics Department, Virginia Polytechnic Institute, Blacksburg, Virginia, 24061

Abstract

Terrell [1989] showed that the probability of a compact polyhedron under a normal distribution could be obtained via Parseval's Theorem and a fast, accurate numerical quadrature. Unfortunately, the efficiency drops with the size of the region, so it is not a particularly good way to obtain tail probabilities, which are often of more practical interest. However, an analogous technique leads to efficient trapezoidal and Gaussian quadratures for normal tails, which actually improve in efficiency as the region moves away from the mean.

I. Introduction

The first demand that statistics makes on numerical computation is for probabilities associated with a normal distribution, since mainstream mathematics has declared that these are not elementary functions in the sense that, for example, a tangent is. Good algorithms are well-known for many such problems; but by no means all. Terrell [1989] shows that one of the more important but difficult cases, that of finding the probability of a polyhedron in a normal multivariate problem, may be well-handled by a class of techniques called Parseval quadratures: The probability is expressed as an integral, which is then transformed to an integral involving the Fourier transforms of functions in the original integral, by an application of Parseval's Theorem. The new integral involves smooth functions on all of space, so that trapezoidal quadratures converge rapidly. Furthermore, one of the factors is a normal density, so that Gauss-Hermite quadrature works very well.

Unfortunately, the method of Terrell [1989] applies only to compact polyhedra, and its efficiency drops rapidly as the maximum distance of the polyhedron from the mean increases. Very often our practical concern is with tail probabilities, applied to regions extending arbitrarily far from the mean. This paper will show how to apply a variant of Parseval quadrature to certain of these probability calculations, in such a way that their efficiency actually increases with the minimum distance of the region from the mean. The paper derives the technique, and then gives examples.

2. An Application of Parseval's Theorem

By a tail probability, we mean in the univariate case for Z standard normal, $\mathrm{P}(Z > z) = Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-Z^2/2} \mathrm{d}Z$ where z>0. Our trick will be first to transform to positive real support by $T = Z - z$. Then

$$Q(z) = \int_0^\infty \frac{1}{\sqrt{2\pi}} e^{-(T+z)^2/2} \mathrm{d}T = e^{-z^2/2} \int_0^\infty \frac{1}{\sqrt{2\pi}} e^{-T^2/2} e^{-zT} \mathrm{d}T.$$

Doubling the integral by reflection about the origin, and expressing the integrand as the product of two densities, we get

$$Q(z) = \frac{1}{z} e^{-z^2/2} \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}} e^{-T^2/2} \frac{z}{2} e^{-|zT|} \mathrm{d}T.$$

Parseval's Theorem says that if f and g are densities, and ϕ and γ their Fourier transforms (characteristic functions of the associated random variables), then $\int fg = \frac{1}{2\pi} \int \varphi\gamma$. This is easy to remember if you think of the Fourier transform as a rotation through a right angle; therefore, it leaves the inner product of two vectors unchanged. Then

$$Q(z) = \frac{1}{z\sqrt{2\pi}} e^{-z^2/2} \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}} e^{-T^2/2} \frac{1}{1 + \left(T/z\right)^2} \mathrm{d}T$$

where I have used the familiar characteristic functions for the normal and Laplace families. We may then compute the integral either by the trapezoidal rule, which will be seen to work well because the integrand is smooth and evaluation points far from zero quickly make negligible contribution. Alternatively, Gauss-Hermite quadrature will turn out to work well, because one of the factors under the integrand is a Gaussian density; and the other factor is smooth and very cheap to compute.

Before we look at these techniques, notice that the Cauchy factor under the integral sign is usually close to one for z large, which is the case of most interest when calculating tail probabilities. Therefore by subtraction we may rewrite

$$Q(z) = \frac{1}{z\sqrt{2\pi}} e^{-z^2/2} \left[1 - \int_{-\infty}^\infty \frac{1}{\sqrt{2\pi}} e^{-T^2/2} \frac{1}{1 + \left(z/T\right)^2} \mathrm{d}T \right].$$

Quadrature in this form will often be convenient, because we are now making a small correction to a classic upper bound for the tail probability.

3. Trapezoidal Quadrature

A trapezoidal rule quadrature formula for the real line is $\int_{-\infty}^\infty f(x) \mathrm{d}x \approx \sum_{i=-\infty}^\infty hf(a + ih)$, where a is a starting point and h is the spacing between the quadrature points. We expect the accuracy to increase for small h; but of course the computational burden increases too. We apply this to the integral in the previous expression, with $a = 0$:

h\z	1.0	2.0	3.0	4.0
0.8	.1590	.02275028	.001349898090	3.1671241856e–5
0.57	.15867	.0227501322	.0013498980316344	3.167124183312e–5
0.4	.15865540	.02275013194820	.0013498980316301	
0.28	.158655254	.0227501319481792		
0.2	.15865525393148			
0.14	.1586552539314571			

Trapezoidal Quadrature of Q(z)

The last entry in each column is accurate to the stated precision. About $9/h$ quadrature points were required to achieve it.

The error for the trapezoidal rule on the line is given by the Poisson summation formula; in the case of Parseval quadrature, that becomes, in our notation,

$$2\sum_{k=1}^{\infty}\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-\frac{2\pi ikt}{h}}\phi(t)\gamma(t)\mathrm{d}t$$

for zero a quadrature point. Notice that we have written it as a sum of inverse Fourier transforms of a product. These are convolutions of the original functions, so that

Theorem: For a trapezoidal Parseval quadrature of $\int fg$ with zero a knot and diameter h, the error is $2\sum_{k=1}^{\infty}\int_{-\infty}^{\infty}f(x)g\left(\frac{2\pi k}{h}-x\right)\mathrm{d}x$.

In our application this becomes $\sum_{k=1}^{\infty}e^{-z^2/2}\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{-T^2/2}e^{-\left|z\left(\frac{2\pi k}{h}-T\right)\right|}\mathrm{d}T$. We notice that this quadrature always overestimates the answer. By contrast, the other symmetric mesh, in which zero is a midpoint, has error the same expression with alternating signs. In particular, the first term is negative. Therefore, in practical cases this other estimate has almost the same size error with opposite signs.

To estimate our error, use the obvious inequality $e^{-|x|}\le e^{-x}$ in the second factor of the integrand. After an integration we get $\text{error}\le\sum_{k=1}^{\infty}e^{-\frac{2\pi kz}{h}}=\frac{1}{e^{\frac{2\pi z}{h}}-1}$. For example, letting $h=0.8$ and $z=1$ we estimate an error .000388; the actual error from our table is .000245. More accurate quadratures give tighter error bounds.

4. Gauss-Hermite Quadrature

Gauss-type quadratures approximate integrals of the form $\int fw$ where w is a positive weight function of finite integral by a sum of the form $\sum_{i=1}^{n}w_i f(x_i)$, where the constant weights w_i and quadrature points x_i are chosen simultaneously so that the quadrature is exact on all polynomials of degree $2n$. In the integral of section 2, the obvious choice of weight is a normal density; in that case quadrature is called Gauss-Hermite quadrature. This is because the points x_i are the roots of the Hermite polynomial of degree n. These are orthogonal with respect to the normal weight function. Then $f(x)=\frac{1}{1+(z/x)^2}$; the cost of each evaluation of this function is amazingly small, and by symmetry we need only $n/2$ of them.

The constants were obtained from Abramowitz and Stegun [1972, p.924], and the following examples evaluated:

n\z	1.0	2.0	3.0	4.0
5	.1675	.022813	.00135020	3.167212e–5
10	.15717	.02274823	.0013498962	3.167124069e–5
exact	.158655	.02275013	.00134989803	3.1671241833e–5

These examples do not exhibit the very high degrees of accuracy in the earlier table; but you must remember that the first row corresponds to only two evaluations of f, and the second to only five!

There exists an error estimate for Gauss-type quadratures, due to Markoff (see e.g. Davis [1975, p.344]); but it seems to be enormously larger than the observed error in all examples tested, and so is of no practical value.

5. Multivariate Normal Tails by Parseval's Theorem

There are well-known, rapid methods for finding univariate normal tail probabilities; but not so for the multivariate case. We will let a multinormal tail region be defined as follows: let $X\approx N(\mathbf{0},\Sigma)$. Then by $\mathbf{X}>z$ we will mean that the random variable exceeds the fixed vector z coordinate by coordinate. Then

$$\mathrm{P}(\mathrm{X}>z)=\int_{\mathrm{X}>z}\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}e^{-\frac{1}{2}\mathrm{X}^{\mathrm{T}}\Sigma^{-1}\mathrm{X}}\mathrm{dX}.$$

We move the corner of our orthant to the origin by the change of variables $\mathbf{Y}=\mathbf{X}-\mathbf{z}$, and expand the exponent to get

$$\mathrm{P}(\mathrm{X}>z)=e^{-\frac{1}{2}z^{\mathrm{T}}\Sigma^{-1}z}\int_{\mathrm{Y}>0}\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}}e^{-\frac{1}{2}\mathrm{Y}^{\mathrm{T}}\Sigma^{-1}\mathrm{Y}}e^{-\mathrm{Y}^{\mathrm{T}}\Sigma^{-1}z}\mathrm{dY}$$

As before, our tactic will be to expand the integrand by reflection to cover Euclidean n-space; then the second factor will be a product of Laplace densities. This only works when we impose a condition on the orthant corner z:

Defi For a multinormal $X \approx N(\mathbf{0},\Sigma)$ random vector, $X > z$ will be called a **tail orthant** whenever $\Sigma^{-1}z > \mathbf{0}$.

This requires that the marginal density along each edge of the region be everywhere decreasing. In this case,

$$P(X>z)=\frac{e^{-\frac{1}{2}z^T\Sigma^{-1}z}}{\prod_{i=1}^{n}\left(\Sigma^{-1}z\right)_i}\int \frac{e^{-\frac{1}{2}Y^T\Sigma^{-1}Y}\prod_{i=1}^{n}\left(\Sigma^{-1}z\right)_i}{(2\pi)^{n/2}|\Sigma|^{1/2}\;2^n}e^{-\sum_{i=1}^{n}|Y_i|\left(\Sigma^{-1}z\right)_i}dY$$

Now apply the multivariate Parseval's theorem to get

$$P(X>z)=\frac{e^{-\frac{1}{2}z^T\Sigma^{-1}z}}{\prod_{i=1}^{n}\left(\Sigma^{-1}z\right)_i(2\pi)^n}\int e^{-\frac{1}{2}t^T\Sigma t}\prod_{i=1}^{n}\left(1+\frac{t_i^2}{\left(\Sigma^{-1}z\right)_i^2}\right)^{-1}dt\;;$$

which has a smooth integrand, well suited to quadrature. The computations become much cheaper if we transform to a spherically-symmetric normal distribution:

$$P(X>z)=\frac{e^{-\frac{1}{2}z^T\Sigma^{-1}z}}{\prod_{i=1}^{n}\left(\Sigma^{-1}z\right)_i(2\pi)^n\det(\Sigma)^{1/2}}\int e^{-\frac{1}{2}t^Tt}\prod_{i=1}^{n}\left[1+\frac{\left(\Sigma^{-1/2}t\right)_i^2}{\left(\Sigma^{-1}z\right)_i^2}\right]^{-1}dt\,.$$

We may compute this by a trapezoidal quadrature on a square mesh of width h. The error analysis parallels the univariate case. For example, let (X,Y) be standard normal variables with correlation 0.5; we compute $P(X>z, Y>z)$ for various values of z:

h\z	1.0	2.0	3.0	4.0
0.57	.05215	.003611099	7.57171105e-5	4.6070648017e–7
0.4	.052294	.00361111817	7.5717111563e-5	4.607064802006e–7
0.28	.05230034	.00361111821034	7.5717111562985e-5	
0.2	.0523004066	.0036111182103472		
0.14	.052300406764267			
0.1	.0523004067642894			

Trapezoidal Quadrature of P(X>z, Y>z)

Here the starting point was not the origin, but $(h/2,h/2)$; therefore the estimates are on the low side. Approximately the square of the number of function evaluations were required here compared to the univariate case; but a Mac Quadra 800 took less than a second to do the longest of these computations.

Gauss Hermite quadrature may be productively applied here; the evaluation points and weights are just the tensor product of those in the univariate case.

6. Bibliography

Abramowitz, A., and Stegun, I. A. (1972). **Handbook of Mathematical Functions.** Dover: New York.

Davis, P. J. (1975). **Interpolation and Approximation.** Dover: New York.

Terrell, G. R. (1989). Parseval quadrature for computing multinormal probabilities. *Technical Report 89-2, Department of Statistics.* Virginia Polytechnic Institute and State University.

COMPUTATION OF MULTIVARIATE NORMAL PROBABILITIES OVER CONVEX REGIONS

Paul N. Somerville
University of Central Florida

Morgan C. Wang
University of Central Florida

ABSTRACT

A method is presented for numerical evaluation of a multivariate normal integral over any convex region. The method is efficient for interactive analyses for moderate accuracies (e.g. approx. two decimal places). The method has been applied to the computation of critical values for multiple comparison methods in the general linear model.

1. INTRODUCTION

Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_k)'$ have the multivariate normal distribution $f(x) = MVN(\mu, \sigma^2\Sigma)$ where Σ is the correlation matrix of $\boldsymbol{x}$ and σ is a scalar. There are many problems in statistics that require computation of $f(x)$ over some region R. That is

$$\int_R f(x)\, dx.$$

For the case when the region of integration is rectangular, the problem has been addressed by many authors. They include Gupta (1964), Milton (1972), Schervish (1984), Deak (1986), Olson and Weissfeld (1991), Genz (1992), Drezner (1992), and Kennedy and Wang (1991, 1992). However, regions of integration for statistical applications such as multiple comparison procedures are not rectangular. For example, the critical value q for $(1 - \alpha)$ simultaneous confidence intervals for Tukey's (1953) pairwise differences of population means can be obtained by solving the following probability equation for q

$$Prob\,\{[(y_i - y_j) - (\mu_i - \mu_j)] \leq q(v_{ij}/2)^{1/2} \text{ for } i \neq j\} = 1 - \alpha$$

where y_i is the least squares estimate of μ_i and v_{ij} is the MVUE for the variance of $y_i - y_j$. The region of integration is convex and bounded by $k(k - 1)$ hyperplanes. In this paper, we describe a method for computation of the multivariate normal integral over any convex region. The region of integration is essentially reduced to a single dimension and integration is accomplished with the assistance of Monte Carlo methods.

2. METHODOLOGY

Although the method is valid for any convex region, we shall assume the region of integration includes the origin. With no loss of generality, we assume the mean is at the origin. Further, in our development, we shall assume the region R is bounded by $m\ (\geq 1)$ hyperplanes and is described by

$$L\boldsymbol{x} \leq \boldsymbol{d}$$

where $L' = (l_1, l_2, \ldots, l_m)$ and the j^{th} hyperplane is given by $l_j'x \leq d_j$. Our first step is to make a transformation so that the new variables $w_1, w_2, \ldots, w_k$ are NID($\mathbf{0}$, I). Let $\Sigma = T\,T'$ (Cholesky decomposition), and set $\boldsymbol{x} = T\,\boldsymbol{w}$. The region R becomes

$$G\boldsymbol{w} \leq \boldsymbol{d}$$

where $G = L\,T$. Setting $G' = (\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_m)$, the j^{th} hyperplane becomes $\boldsymbol{g}_j'w = d_j$.

We discuss two cases.

Case 1 Σ, σ^2 known

Case 2 Σ known, σ^2 unknown, s^2 is an unbiased estimate of σ^2 with ν degrees of freedom.

Our strategy is as follows.

a) Choose a unit random direction $\mathbf{c} = (c_1, c_2, \ldots, c_k)$.

b) Obtain distance r to the boundary in the direction $\mathbf{c}$.

Case 1.

c) Since $r^2 = \boldsymbol{w'w}$, $r^2 = \boldsymbol{c'c}$ has a χ^2distribution with k degrees of freedom.

d) $Prob[\chi^2 \leq r^2]$ is an unbiased estimate of the integral value.

Case 2.

c) $(r^2/k)/s^2$ has $F_{k,\nu}$ distribution.

d) $Prob[F_{k,\nu} \leq (r^2/k)/s^2]$ is an unbiased estimate of the integral value.

e) Repeat steps a) to d) until the average of the estimates has a specified standard error.

3. DISTANCE TO THE BOUNDARY

For a given direction $\boldsymbol{c}$, the distance from the origin to the plane j is $d_j /(\boldsymbol{g}_j'\boldsymbol{c}) = d_j /a_j = r_j$ (say). The intersection of the line with the plane j is $\boldsymbol{w} = d_j/(\boldsymbol{g}_j'\boldsymbol{c})\ \boldsymbol{c} = r_j \boldsymbol{c}$. The above intersection will be in the region R if

$$\boldsymbol{g}_j'\boldsymbol{w} = \boldsymbol{g}_j' r_j \boldsymbol{c} = r_j a_i \leq d_i \quad i \neq j = 1,2, ...,m.$$

The distance to the boundary in the direction $\boldsymbol{c}$ is the distance to the plane for which $r_j a_i \leq d_i$ for all $i \neq j$.

4. ALGORITHM

1. Input data: Σ, σ^2 *or* s^2 and ν, L, $\boldsymbol{d}$, m, k, ε, $NMAX$ and $SEED$.

2. Obtain the matrix $G = L\ T$, where T is the lower triangular matrix of the Cholesky decomposition for σ^2 or $\sigma^2\Sigma$.

3. Set $SUM = SUMSQ = 0, N = 0, STD = 0$.

```
Do while $STD < \varepsilon$ and $N < NMAX$
    Set $TSUM = 0$ and $TSUMSQ = 0$
    Repeat (a) to (c) $10k^2$ times
        a) Generate a unit random direction $\boldsymbol{c}$.

        b) If for some $j$, $r_j a_i \leq d_i$ for all $i \neq j = 1, ... ,m$,
        where $a_j = \boldsymbol{g}_j'\boldsymbol{c}$, $r_j = d_j/a_j$,
        then $r = r_j$, set $tt = Prob[\chi_k{}^2 \leq r^2]$ or
        $tt = Prob[F_{k,n} \leq (r^2/k)/s^2]$,
        else set $tt = 1$.

        c) $TSUM = TSUM + tt$,
        $TSUMSQ = TSUMSQ + tt*tt$.

        d) $N = N + 1$,  $SUM = SUM + TSUM$,
        $SUMSQ = SUMSQ + TSUMSQ$,
        $MVN = SUM /10Nk^2$,
        $STD = ((SUMSQ - SUM*MVN)/(10Nk^2$
        $(10Nk^2 - 1)))^{1/2}$.

        4. Output is $MVN$, $STD$, and $10Nk^2$.
```

5. EXPERIMENTAL RESULTS

The following integral was calculated for 40 different sets of parameters.

$$\int...\int MVN(0,\Sigma)\ dx$$

Σ was a correlation matrix with equal non-diagonal elements ρ, and the limits of integration for each variable were -∞ to a. Values for ρ were 0 (.1) .9 and four diferent values of a were randomly chosen from the interval (1.5,2.5). Sufficient random directions were obtained to obtain a standard error of .002 for the calculated value of the integral. The number of random directions required was approximately 100 k^2. The following table gives the average absolute error and the standard deviation of the average absolute error for various values of k.

k	average absolute error	sd of average absolute error
3	.0016	.0014
4	.0018	.0015
5	.0013	.0011
6	.0022	.0019
7	.0016	.0013
8	.0018	.0014
9	.0019	.0015
10	.0021	.0018
12	.0011	.0007
14	.0016	.0014
16	.0013	.0010
20	.0013	.0011

6. CONCLUSIONS

A method is presented for the evaluation of a multivariate normal integral over any convex region. The method is efficient for moderate acuracies (e.g. approximately two decimal places. Quoting from Berger (1991), "...for statistical problems ... two significant digit accuracy typically suffices, and only rarely are more than three ... needed." The method is thus practical for a wide variety of statistical problems. A typical application is to problems in multiple comparisons. Application of the methods presented here are given in Somerville (1993a,b, 1994). Computation times are such that interactive statistical analyses are practical on 80386 or 80486 processors.

Future research on more sophisticated sampling and computational methods should significantly decrease processing times. The method is well adapted to the use of parallel processors.

7. REFERENCES

Berger, James (1991), "Introduction", Contemporary Mathematics, **115**, 1-7.

Deak, I (1986), "Computing probabilities of rectangles in case of multinormal distribution", Journal of Statistical Computation and Simulation, **26**, 101-114.

Drezner, Z. (1992), "Computation of multivariate normal integral" ACM Transactions on Mathematical Software, **18**, No. 4, 470-480.

Genz, A. (1992), "Numerical computation of multivariate normal probabilities", Journal of Computational and Graphical Statistics, **1**, 141-149.

Gupta, S.S. (1964), "Probability integrals of multivariate normal and multivariate t", Annals of Mathematical Statistics, **34**, 792-838.

Milton, R.C. (1972) "Computer evaluation of the multivariate normal integral", Technometrics **14**, 881-889.

Olson, J.M. and Weissfeld, L.A. (1991). "Approximation of certain multivariate integrals", Statistics and Probability Letters, **11**, 309-317.

Schervish, M. (1984), "Multivariate normal probabilities with error bound", Applied Statistics, **33**, No. **1**, 81-87.

Somerville. P.N. (1993a) "Exact all-pairwise multiple comparisons for the general linear model", Proceedings of the 25th Symposium on the Interface, Computing Science and Statistics, April 1993 352-356.

Somerville. P.N. (1993b) "Simultaneous multiple orderings", Technical Report 93-TR-1 Department of Statistics, University of Central Florida, June 1993.

Somerville, P.N. (1994) "Multiple Comparisons", Technical Report 94-TR-1 Department of Statistics, University of Central Florida, May 1994.

Tukey, J.W. (1953), " The problem of multiple comparisons" Mimeographed manuscript, Princeton University.

Wang, M.C. and Kennedy, W.J. (1990), Comparison of algorithms for bivariate normal probabilities over a rectangle based on self-validating result from interval analysis", Journal of Statistical Computation and Simulation, **37**, 13-25.

Wang, M.C. and Kennedy, W. J. (1992), "A numerical method for accurately approximating multivariate normal probabilities", Computational Statistics and Data Analysis, **13**, 197-210.

Wang, M.C. and Kennedy, W.J. (1994), "Self-validating computations of probabablities and percentiles for selected central and non-central univariate probability functions", Journal of American Statistical Association, **89**, 1000 - 1010.

Efficient Programs for Simulating Chi-bar Square Distributions

Shixian Qian
Carnegie Mellon University

June 27, 1994

Abstract

There is no program for doing order restricted statistical inference in any major statistical software package due to the hurdle of computation. As a first step, we now develop a set of efficient programs for simulating p-values of chi-bar square statistics and level probabilities, which includes complete order, matrix order, cubic order, tree order and general partial order. We discuss efficiency and accuracy of these programs, compare two methods of simulating p-values for chi-bar square statistics, (1) a direct method and (2)computing p-values by Monte Carlo estimates of the level probabilities. We give an example of a clinical trial to illustrate the applications of these programs.

1 Introduction

The $\overline{\chi}^2$ distributions are the most important kind of distributions in order restricted statistical inference. The book ≪Order restricted statistical inference≫ written by Robertson, Wright and Dykstra (1988) studied $\overline{\chi}^2$ distributions in Chapters two and three. Because of the variety and complexity of partial orders, it is difficult to study $\overline{\chi}^2$ distributions for general partial orders from their point of view. Therefore, the book only deals with several simple cases, such as a complete order or a simple tree order on an index set with small size. This situation made application of order restricted inference be very restrictive, none of major statistical software packages contains $\overline{\chi}^2$ distributions and order restricted inference. Because of development of efficient algorithms for isotonic regressions and modern computers, we are able to study $\overline{\chi}^2$ distributions from application point of view with computer intensive method. In this article, we provide two methods, RF(relative frequency) method and LP(level probability) method, to obtain p-values of $\overline{\chi}^2$ distributions, and compare their accuracy.

For simplicity, we first introduce isotonic regressions on a two dimensional grid. Let $X = \{(i,j) : i = 1,\dots,I; j = 1,\dots,J\}$ be an $I \times J$ grid. A function $f(\cdot,\cdot)$ on X is said to be isotonic if $f(\cdot,\cdot)$ is increasing in both variables. A function $g^*(\cdot,\cdot)$ on X is said to be an isotonic regression of a given function $g(\cdot,\cdot)$ with known weights $w(\cdot,\cdot)$ on X, if $g^*(\cdot,\cdot)$ is a solution of the following minimization problem:

$$\min_f \sum_{i,j}(g(i,j) - f(i,j))^2 w(i,j)$$

subject to $f(\cdot,\cdot)$ is isotonic on X.

Let $g(i,j), i = 1, ..., I; j = 1, ..., J$ be $n = I \times J$ independent normal random variables with mean 0 and variance $1/w(i,j)$, $\overline{g}$ be their weighted sample mean, and $g^*(\cdot,\cdot)$ be the isotonic regression of $g(\cdot,\cdot)$ with weights $w(\cdot,\cdot)$. Then $X_{01} = \sum_{i,j}(g^*(i,j) - \overline{g})^2 w(i,j)$ is called a $\overline{\chi}^2_{01}$ random variable; and $X_{12} = \sum_{i,j}(g^*(i,j) - g(i,j))^2 w(i,j)$ is called a $\overline{\chi}^2_{12}$ random variable. Let M be the number of distinct values of the isotonic regression $g^*(\cdot,\cdot)$, then M is a random variable taken values $1, \ldots, n$. The probability distribution of M is called level probabilities. $\overline{\chi}^2$ distributions are mixture of χ^2 distributions. Rebertson et al(1988) shows that,

$$P(X_{01} > c) = \sum_{k=2}^{n} P(M = k)P(\chi^2_{k-1} > c); \quad (1)$$

$$P(X_{12} > c) = \sum_{k=1}^{n-1} P(M = k)P(\chi^2_{n-k} > c). \quad (2)$$

The definitions of isotonic regression, $\overline{\chi}^2$ distributions and level probabilities can be generalized to a partially ordered finite index set X.

2 An Example

Cornfield (1962) provides a data set from a clinical trial. It is also listed in Agresti(1990). This is a sample of male residents of Framingham, Massachusettes, aged 40-59 classified both by blood pressure and serum cholestrerol level. In this data set, the response variable is a binary variable with 1 for occurrence of heart disease and 0 for non-occurrence. By medical theory, the rate of presenting heart disease increases when blood pressure(x) or serum cholesterol(y) increases. The covariate x is divided into 8 ordered levels, and the covariate y is divided into 7 ordered levels. Therefore, we have 56 cells in total. The cell sample proportions can be viewed as a function $g(\cdot,\cdot)$ on an 8×7 grid. The ordered maximum likelihood estimate function $g^*(\cdot,\cdot)$ of the rate of presenting heart disease, which is increasing both in blood pressure and serum cholesterol, is an isotonic regression of $g(\cdot,\cdot)$ with given weights $w(\cdot,\cdot)$, where $w(i,j)$ is the number of observations in the (i,j) cell. The ordered maximum likelihood estimates are listed in Table 1.

In order to verify the statement that the rate of presenting heart disease increases when blood pressure or serum cholestrerol increases, we consider following models for the data set.

M_0: constant rate model, that is, the rate of heart disease depends on neither blood pressure nor serum cholestrerol.

M_1: isotonic rate model, that is, the rate of heart disease is increasing both on blood pressure and serum cholestrerol.

M_2: saturated model, that is, the rate of heart disease is arbitrary.

Let T_{01} be a likelihood ratio test statistic for testing independence model M_0 vs isotonic model $M_1 - M_0$. Let T_{12} be a likelihood ratio test statistic for testing M_1 vs saturated model $M_2 - M_1$. Computational formulas for T_{01} and T_{12} can be found in Robertson et al(1988). For the Cornfield data set, $T_{01} = 70.66$ and $T_{12} = 50.50$. Robertson and Wegman (1978) has shown that, the asymptotic distributions of the likelihood ratio test statistics T_{01} and T_{12} are $\overline{\chi}^2_{01}$ and $\overline{\chi}^2_{12}$ distributions respectively.

3 P-values

There are two ways to obtain p-values of $\overline{\chi}^2$ distributions by computer intensive method. The first one is a direct method. We generate a random sample of a $\overline{\chi}^2$ distribution with size of N, count the frequency f of which

Table 1: Ordered Maximum Likelihood Estimates for the Cornfield Data

	-200	200-209	210-219	220-244	245-259	260-284	284-
-117	0.0106	0.0106	0.0106	0.0106	0.0106	0.0303	0.0303
117-126	0.0106	0.0250	0.0364	0.0478	0.0478	0.0990	0.0990
127-136	0.0242	0.0250	0.0364	0.0478	0.0478	0.0990	0.0990
137-146	0.0242	0.0250	0.0364	0.0741	0.0943	0.0990	0.1618
147-156	0.0364	0.0364	0.0364	0.0845	0.0943	0.1618	0.1618
157-166	0.0364	0.0364	0.0364	0.0845	0.0943	0.1618	0.2679
167-186	0.0811	0.0811	0.0811	0.0845	0.2500	0.2679	0.2679
186-	0.1667	0.1667	0.2500	0.2500	0.2500	0.2679	0.2679

exceed the specified test statistic c, then the relative frequency $\hat{p} = f/N$ is an unbiased estimate of $p = P(\overline{\chi}^2 > c)$. We call this method RF(relative frequency) method. In RF method, $N\hat{p}$ is a binomial random variable with mean Np and variance $Np(1-p)$. Thus, $\hat{p}$ has mean p and variance $p(1-p)/N$. The second method is using simulation to obtain estimates of level probabilities, then applying formula (1) or (2) to get an estimate $\tilde{p}$ of p. We call this method LP(level probability) method. Let π_k be an unbiased estimate of $P(M = k)$, for $k = 1, \ldots, n$. Then a LP estimate $\tilde{p}$ of p can be obtained by following formulas.

$$\tilde{p} = \sum_{k=2}^{n} \pi_k P(\chi^2_{k-1} > c), \qquad (3)$$

$$\text{or } \tilde{p} = \sum_{k=1}^{n-1} \pi_k P(\chi^2_{n-k} > c). \qquad (4)$$

The LP estimate $\tilde{p}$ is also an unbiased, consistent estimate of p. Applying results in section 12.1.5 of Agresti(1990), we obtain following theorem.

Theorem 1 *(1).* $E(\tilde{p}) = p$.
(2a). For $\overline{\chi}^2_{01}$ *random variable,* $Var(\tilde{p}) = [\sum_{k=2}^{n} P(\chi^2_{k-1} > c)^2 P(M = k) - p^2]/N;$
(2b). For $\overline{\chi}^2_{12}$ *random variable,* $Var(\tilde{p}) = [\sum_{k=2}^{n} P(\chi^2_{n-k} > c)^2 P(M = k) - p^2]/N;$
(3).$Var(\tilde{p}) \leq Var(\hat{p})$*.*

Thus, We can estimate variance of $\tilde{p}$ by following formulas.

$$Va\hat{r}(\tilde{p}) = [\sum_{k=2}^{n} P(\chi^2_{k-1} > c)^2 \pi_k - \tilde{p}^2]/N; \quad (5)$$

$$Va\hat{r}(\tilde{p}) = [\sum_{k=1}^{n-1} P(\chi^2_{n-k} > c)^2 \pi_k - \tilde{p}^2]/N; \quad (6)$$

4 Programs

Programs we developed for $\overline{\chi}^2$ distributions are based on algorithms described in Eddy and Qian(1994) and Qian(1992), which are shown to be better than other algorithms for isotonic regressions. The *xbarg* program for p-values of $\overline{\chi}^2$ distributions on an arbitrary partial ordered set, which is written in C language, applies acceptance sampling method to generate normal random variables, uses the IBCR algorithm in Qian(1994) to find isotonic regressions, and utilizes recursive formula to find p-values of χ^2 distributions. This program requires to input a partial order. To simplify the input and make the program

Table 2: Time(in seconds) of the programs

	size	time
xbarl	1000	930.87
xbart	15	16.34
xbar2	20 × 20	6003.75
xbar3	3 × 3 × 3	210.99
xbarg	30	262.07

Table 3: P-values for $\bar{\chi}^2$ statistics in Cornfield(1962) Example

	$T_{01} = 70.66$		$T_{12} = 50.50$	
	p	stdev	p	stdev
RF	0.000	0.0000	0.394	0.0015
LP	0.000	0.0000	0.393	0.0003

more efficient, we implement other four programs, *xbarl*, *xbart*, *xbar*2 and *xbar*3 programs for simulating p-values of $\bar{\chi}^2$ distributions on a completely ordered set, a tree ordered set, a rectangular grid with componentwise increasing order, and a cubic grid with componentwise increasing order respectively. These five programs are both for equal and unequal non-negative weights.

Utilizing results in Eddy and Qian(1994) and Qian(1992), we find that *xbarl*, *xbart*, and *xbar*2 are strongly ploynomial time programs. Qian(1992) also gives worst time complexity of *xbar*3 and *xbarg* programs. Table 2 lists the time(in seconds) to run these programs on an HP 9000 model 735 workstation with 100,000 simulations.

We use *xbar*2 program to find p-values of $\bar{\chi}^2$ statistics in Cornfield example with 100,000 simulations on a 486 DX/50 PC. It took 1159 seconds to obtain p-values of these $\bar{\chi}^2$ statistics. Results are listed in Table 3.

The p value of T_{01} is 0.0, so we reject independence model M_0. The p value of T_{12} is 0.393, so there is no significant difference between isotonic model M_1 and saturated model M_2. Thus, we prefer isotonic model $M_1 - M_0$, in other words, we have statistical evidence that the rate of heart disease increases when blood pressure or serum cholestrerol increases. The Table 1 gives estimates of the rate of heart disease in different levels of blood pressure and serum cholestrerol.

5 Accuracy

The accuracy of the results is the most important issue for simulations. It depends on good quasi-random number generators. We use several ways to test our normal random number generators.

We use tables in the appendix of Robertson et al(1988) to check our programs. In Tables 1-5 in their appendix, excluding boundary points, there are 99 cases for $\bar{\chi}^2_{01}$ distributions and 99 cases for $\bar{\chi}^2_{12}$ distributions, each case has 6 critical values which have p-values 0.1, 0.05, 0.025, 0.01, 0.005 and 0.001 respectively. We calculated all the p-values for these critical values by both RF method and LP method. Among 1188 RF estimates $\hat{p}$ we calculated, there are 44 estimates that the true p-values are not within two standard deviations of its estimate $\hat{p}$. The failure rate is about 3.7%. For LP estimate $\tilde{p}$ of p, we say a case is failure, if one of the true values is not within two standard deviations of its estimate $\tilde{p}$. We find there are 5 failures among 99 cases for $\bar{\chi}^2_{01}$ distributions, and 4 failures among 99 cases for $\bar{\chi}^2_{12}$ distributions. The failure rate is approximately 4.5%. The results are agreed with theoretical analysis. We computed level probabilities for complete orders, simple tree orders and unimodel orders, and compared results with Tables 10, 11 and 20 in the appendix of Robertson et al(1988). We also computed level probabilities for rectangular grids and compared results of Moonesinghe and Wright(1994). They are all very close.

Then , we apply theoretical results in Section 3 to do more checks. Let $p = P(\overline{\chi}^2_{01} > c)$. First, we can use different seeds to get a random sample of p, so we can obtain sample estimate and sample standard deviation(SSD). Clearly SSD should be almost same as the standard deviation by the formulas listed in Section 3. Second, If the numbers of simulations are different, then the ratio of the standard deviation estimates should be the square root of the ratio of the numbers of simulations. Third, the LP estimate $\tilde{p}$ is better than the RF estimate $\hat{p}$. Actually what we did is the following. In Cornfied data example, we choose six values for $\overline{\chi}^2_{01}$ statistics, and six values for $\overline{\chi}^2_{12}$ statistics. we get estimates of p by 1000 simulations, 10,000 simulations and 100,000 simulations respectively. We repeated this procedure 100 times. Table 4 lists statistics for $p = P(\overline{\chi}^2_{01} > 12.571289)$. Due to the length of this paper, the other eleven tables are omitted. Our simulations show that, the SSDs are almost the same as the one calculated by (5); the ratios of standard deviations are near $\sqrt{0.1} = 0.316228$; the LP estimate $\tilde{p}$ is statistically better than the RF estimate $\hat{p}$. We applied this method for complete orders, tree orders and special partial orders with equal weights or unequal weights and got same conclusions. So we are sure our normal random generator works really well. From our extensive simulations, we have following recommandations for simulating $\overline{\chi}^2$ distributions.

1. Use LP estimate $\tilde{p}$ instead of $\hat{p}$.
2. Apply formulas (5) and (6) to get an estimate of the standard deviation of $\tilde{p}$.
3. In hypothesis testing, do 1000 simulations first, then run 100,000 simulations if necessary.

Table 4: Some Statistics for $p = P(\overline{\chi}^2_{01} > 12.571289)$ in Cornfield Example

	1000	10000	100000
$\tilde{p}$	0.102418	0.101287	0.099986
stdev	0.003269	0.001065	0.000333
ratio		0.325788	0.312676
SSD	0.00347	0.00108	0.00031
$\hat{p}$	0.093	0.0989	0.10152
stdev	0.009184	0.002985	0.000955
ratio		0.325022	0.319933
SSD	0.00936	0.00308	0.00096

References

[1] Agresti,A.(1990)*Categorical Data Analysis*. John Wiley & Sons, New York.

[2] Cornfield, J. (1962). Joint dependence of risk of coronary heart disease on serum cholestrerol and systolic blood pressure: a discriminant function analysis. *Fed. Proc. 21 Supplement No. 11*, 58-61.

[3] Eddy, W. and Qian, S.(1994) An algorithm for isotonic regression with two covariates. (in preparing)

[4] Moonesinghe, R. and Wright, F.T.(1994) Likelihood ratio tests involving a bivariate trend in two-factor designs: the level probabilities. Technical Report, Department of Statistics, University of Missouri.

[5] Qian, S.(1992) Algorithms for isotonic regressions and related theory. PH. D Thesis. Dept. of Mathematics & Statistics, University of Pittsburgh.

[6] Robertson, T., Wright, F.T. and Dykstra, R.L. (1988). *Order restricted statistical inference*. John Wiley & Sons, Chichester.

The Cumulative Score Control Chart for an Open Loop Control

Y. Eric Shao

Department of Statistics, College of Management
Fu Jen University
Taipei, Taiwan, R.O.C.

Abstract

Recently, interest concerning the utilization of statistical process control (SPC) and engineering process control (EPC) has increased. This research is concerned with the utilization of SPC techniques in process control. For simplicity, this research considers the case where no feedback control action is in a process: an open loop control. In order to explain the utilization of the cuscore control charts, two cases -- one case's underlying process follows AR(1) and the other follows ARMA(1,1) process with mean shifts in the process -- will be discussed. In addition, some simulation studies will be presented.

Introduction

Recently, interest concerning the utilization of statistical process control (SPC) and engineering process control (EPC) has increased. This research is concerned with the utilization of SPC techniques in process control.

In this research, the main objective of using the SPC techniques is to detect the mean shift or transient disturbance in real time. Since the traditional Shewhart control charts are not sensitive in detecting small shifts in the process [2] [4], the Shewhart control charts were not used in this research. Although the cumulative-sum (or cusum) control chart is effective in detecting small shifts in the process, the cusum chart would be very slow to detect large process shifts [2] [4]. Therefore, this research will not consider cusum charts to detect the unplanned transient disturbances. Instead, since the cuscore chart is not only effective in detecting the small shifts in the process but also provides cuscore statistic which is helpful to identify the unplanned transient disturbances, the cumulative score (or cuscore) chart [1] is appealed in this study.

Cuscore Charts For Open Loop Control

Shao et al. [3] have addressed the control of transient disturbance. For simplicity, this research considers the case where no feedback control action is in a process: an open loop control. In order to explain the utilization of the cuscore control charts, two cases -- one case's underlying process follows AR(1) and the other follows ARMA(1,1) process -- will be discussed.

The details of the cuscore chart can be referenced in [1]. The concept of the cuscore charts is described as follows. Consider a model which can be written as

$$a_t-f(y_t,X_t,m),\quad t-1,2,...,n \qquad (1)$$

where y_t is the output observations, x_t is the independent variable, m is a certain unknown parameter, and f() is some certain function. If m is the true value of the unknown parameter, the resulting a_t's would follow a white noise sequence. Apart from a constant, the log likelihood for m=m$_0$ is

$$l(m) = -\frac{1}{2\sigma^2}\sum_{t=1}^{n} a_{t0}^2$$

where a$_{t0}$^{2}'s are obtained by setting m=m$_0$ in Equation (1). Let

$$-\frac{\partial a_t}{\partial m}\Big|_{m=m_0} = g_{t0} \ , \qquad (2)$$

then the following relationship holds:

$$\frac{\partial l(m)}{\partial m} = \frac{1}{\sigma^2}\sum_{t=1}^{n} a_{t0}\, g_{t0} \ .$$

The cuscore statistics with the parameter value m=m$_0$ is defined as:

$$Q = \sum_{t=1}^{n} a_{t0}\, g_{t0} \ . \qquad (3)$$

Box and Ramirez [1] addressed that the following relationship holds if the model is linear in parameter m and approximate otherwise:

$$a_{t0} = (m-m_0)\, g_{t0}+a_t \ . \qquad (4)$$

Furthermore, Box and Ramirez [1] remarked the so called centred cuscore (CC) is the cuscore evaluated at $m = \overline{m} = (m_0+m_1)/2$, and the CC is defined as:

$$CC = \sum_{t=1}^{n} \overline{a_t}\ \overline{g_t} \ ,$$

where $\overline{a_t} = a_t(\overline{m})$ *and* $\overline{g_t} = g_t(\overline{m})$. This CC is used to signal the out of control situation, and the details of CC can be referenced in [1].

AR(1) Process With Mean Shifts And Cuscore Charts

Consider the case in which the underlying process can be modelled as an AR(1) process; that is, the process follows

$$y_t = \phi y_{t-1}+a_t \ ,$$

where y$_t$ is the output deviation from target at time t, ϕ is a certain parameter, and a$_t$'s stand for white noise.

Now consider the utilization of cuscore control chart to detect the mean shift. Suppose that one want to detect the mean shift which has the magnitude of one standard deviation (i.e., for simplicity, this research considers that one standard deviation equals 1), then the underlying process can be reformed as:

$$y_t = m\delta+\phi y_{t-1}+a_t \ , \qquad (5)$$

where m is a certain parameter, and δ stands for the magnitude of the mean shift. If the process is operated correctly, the parameter m should be zero and there should be no mean shift in the process. On the other hand, if some disturbances exist in the process, the mean shift would be present in the process. Therefore, in this case, the utilization of cuscore chart is equivalent to doing a hypothesis test for the parameter m; that is,

testing m=m$_1$=(1-ϕ)
m=m$_0$=0.

Since this study wants to detect the mean shift with one standard deviation, the δ is set to be 1 and m$_1$ is set to be (1-ϕ). The reason why this study sets m$_1$ equal (1-ϕ) is because of the following fact: $u = \frac{m\delta}{(1-\phi)}$ with the substitution of u and δ with 1.

By using Equations (2), (3), and (4), one is able to obtain the cuscore statistics:

$$Q = \sum_{t=1}^{n} (y_t-\phi y_{t-1})\,\delta \ . \qquad (6)$$

Since one step ahead prediction for AR(1) process is $\hat{y}_T = \phi y_{t-1}$, Equation (6) can be reformed as:

$$Q = \sum_{t=1}^{n} (y_t-\hat{y}_t) \ \ (since\ \delta = 1) \ . \qquad (7)$$

Therefore, by viewing Equation (7), one knows that the utilization of cuscore statistics is equivalent to using cusum statistics on the residuals.

ARMA(1,1) Process With Mean Shifts And Cuscore Charts

Consider the case in which the underlying process can be modelled as a ARMA(1,1) process; that is, the process follows

$$y_t = \phi y_{t-1}+a_t-\theta a_{t-1} , \qquad (8)$$

where y_{t-1} is output deviation from the target at time t, ϕ and θ are certain parameters, and a_t's stand for white noise.

Now consider the utilization of cuscore control chart to detect the mean shift. Suppose that one wants to detect the mean shift which has the magnitude of one standard deviation (i.e., again, this study considers that one standard deviation equals 1), then the underlying process can be reformed as

$$y_t = m\delta+\phi y_{t-1}+a_t-\theta a_{t-1} , \qquad (9)$$

where m stands for some unknown parameter and δ stands for the magnitude of the mean shift. If the process is operated correctly, the parameter m should be zero and there should be no mean shift in the process. On the other hand, if some disturbances exist in the process, the mean shift would be present in the process. Therefore, the utilization of the cuscore chart is equivalent to doing a hypothesis test for the parameter m; that is,

testing $m=m_1=(1-\phi)$
$m=m_0=0$.

Again, since one wants to detect the mean shift with one standard deviation, δ is set to be 1 and m_1 is set to be $(1-\phi)$. The reason why m_1 equals $(1-\phi)$ is same as previous case.

By using Equation (2), (3), and (4), one is able to obtain the cuscore statistics:

$$Q_t = 2\theta Q_{t-1}-\theta^2 Q_{t-2}+ \sum_{t=1}^{n} (y_t-\phi y_{t-1})\delta . \qquad (10)$$

Since in this study one wants to detect the mean shift with one standard deviation, δ is set to be 1, $m_0=0$, and $m_1=(1-\phi)$. Thus,

$$\overline{m} = \frac{m_0+m_1}{2} = \frac{(1-\phi)}{2} ,$$

$$\overline{a_t} = \frac{y_t-(\frac{1-\phi}{2})\delta-\phi y_{t-1}}{(1-\theta B)} ,$$

and $\overline{g_t} = \frac{\delta}{(1-\theta B)}$.

The centred cuscore then would be:

$$CC_t = 2\theta CC_{t-1}-\theta^2 CC_{t-2}+ \sum_{t=1}^{n} [y_t-(\frac{1-\phi}{2})\delta-\phi y_{t-1})\delta .$$

Simulation Studies

To show the utilization of the cuscore statistics, this research examines a simulation study. Assume that the underlying process can be modelled as an ARMA(1,1) process. This study uses Equation (8) to represent a certain process, and a white noise sequence which has mean of zero and a standard deviation of 1 is generated for this certain process. Since this study wants to detect the mean shift with magnitude of one standard deviation, this study shifts the process mean from 0 to 1 after observation 25. In addition, this study arbitrarily chooses $\phi=0.7$ and $\theta=0.6$.

Figures 1, 2, 3, and 4 display the process outputs with no mean shift, residuals of process output with no mean shift, process outputs with mean shift 1 starting at observation 26, and residuals of process output with mean shift 1 starting at observation 26, respectively. Notice that the prediction of the ARMA(1,1) process is

$$\overline{y_t} = \phi y_{t-1}-\theta(y_{t-1}-\overline{y_{t-1}}) .$$

Therefore, the residual of the ARMA(1,1) is calculated as follows:

$$Residual_t = y_t-\overline{y_t} .$$

Figure 1, in fact, represents an ARMA(1,1) process which has the parameters $\phi=0.7$ and $\theta=0.6$ and has a white noise sequence which has mean of zero and a standard deviation of 1. Since there is no mean shift in the process, the residuals plot should behave like a random noise. This characteristic can be seen in Figure 2. Figure 3 represents an ARMA(1,1) process with mean shift of 1 starting at observation 26. Furthermore, since there is a mean shift in the process, the residuals plot should not behave like a random noise. This characteristic can be observed in Figure 4.

Figure 5 shows the plot of cuscore statistics, centred cuscore statistics, and summation of the residuals of process output when the ARMA(1,1) process has mean shift of 1 starting at observation 26. Since in this study the process has no mean shift occurring before observation 26, one can expect that the values of cuscore statistics and the summation of the residuals are around zero before observation 26. Also, since the process has a mean shift of 1 starting at observation 26, one can expect that the values of cuscore statistics and the summation of the residuals are positively increased over time after observation 26 (i.e., since the mean shift is a positive value.) In addition, since the centred cuscore is used to signal the out of control situation, one can expect that the slope of the centred cuscore would be changed some certain time around (or after) observation 26. These characteristics are all shown in Figure 5.

Furthermore, one can observe that the minimum point of centred cuscore statistics line is at observation 25. The meaning of the minimum point is that the slope of the centred cuscore line is changed. One should take the difference between the value of the centred cuscore at time t (i.e., the time after the occurring minimum point) and this minimum point to determine whether the out of control signal is given or not. For example, in this simulation study, the minimum point, which is equal to -20.587, occurs at observation 25. Therefore, one can expect that the out of control signal is given at observation 34 since the value of the difference between cuscore statistics (at observation 16) and the minimum point is 15.62, and this value is greater than the boundary h, 15.35. The boundary h is defined as [1]:

$$h = \frac{[\sigma^2 \ln(\frac{1}{\alpha})]}{(m_1 - m_0)}$$

Therefore, h=15.35 if one chooses α=0.01.

Summary

This study is concerned with the utilization of cuscore control chart in an open loop process. The concept of using cuscore control chart is discussed for two open loop processes, AR(1) and ARMA(1,1). In addition, the simulation study demonstrates how to use the cuscore control chart to detect the mean shift in the process.

The objective of this paper is to show that the cuscore control chart is useful in detecting the mean shift or transient disturbance. Ongoing research is developing a technique for detecting the mean shift or transient disturbance in real time for a closed loop process.

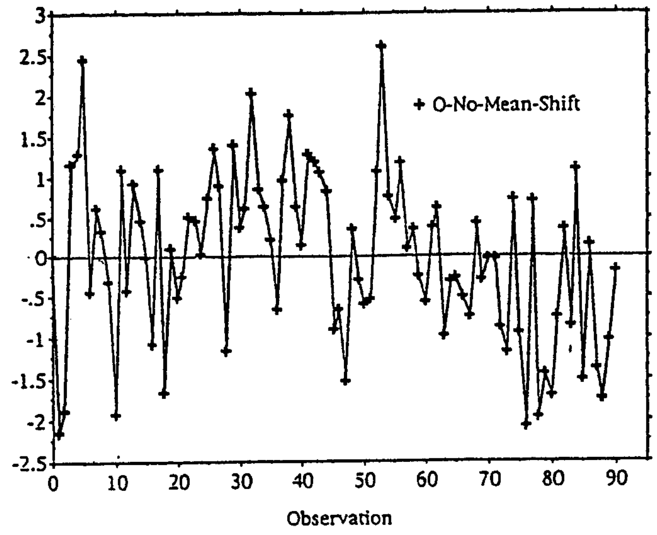


Figure 1:
Plot of output deviations from target which are generated by an ARMA(1,1) process with no mean shift

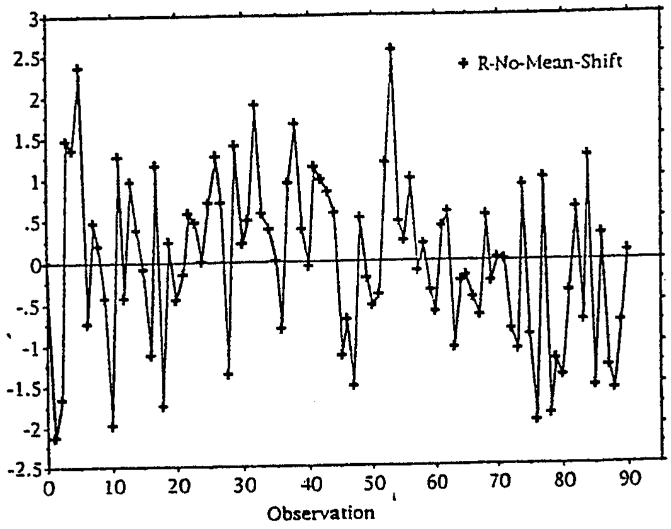


Figure 2:
Plot of residuals which are generated by an ARMA(1,1) process with no mean shift

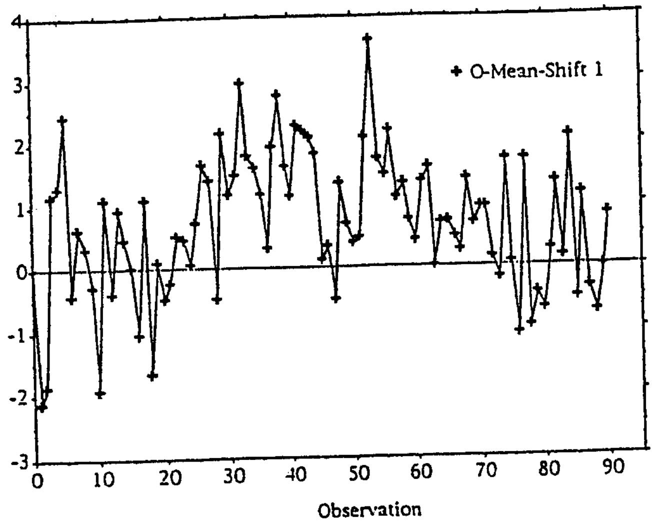


Figure 3:
Plot of output deviations from target which are generated by an ARMA(1,1) process with mean shift 1 starting at observation 26

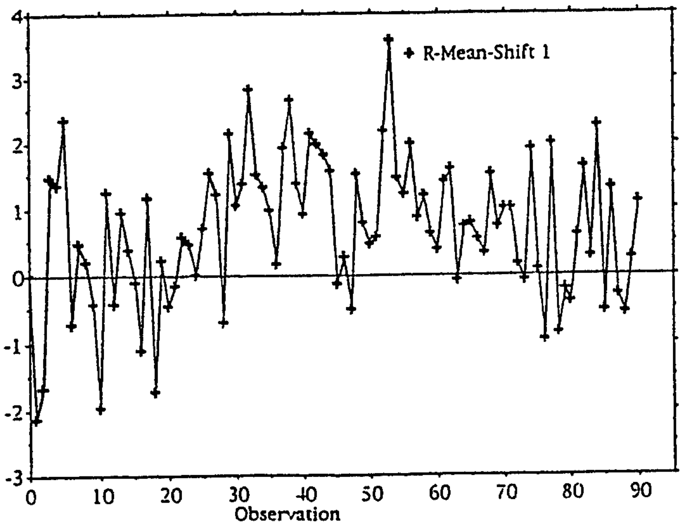


Figure 4:
Plot of residuals which are generated by an ARMA(1,1) process with mean shift 1 starting at observation 26

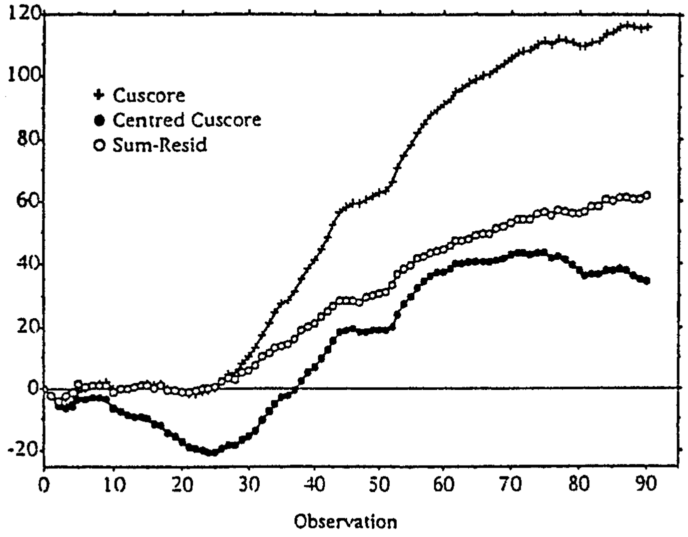


Figure 5:
Plot of cuscore, centred cuscore, and summation of the residuals of process output when the ARMA (1,1) process has mean shift 1 starting at observation 26

References

[1] Box, G. and Ramirez, J. (1992), "Cumulative Score Charts," *Quality and Reliability Engineering International*, Vol. 10, pp. 17-27.

[2] Montgomery, D.C. (1991). *Introduction to Statistical Quality Control* (2nd Edition). John Wiley & Sons, Inc., New York, NY.

[3] Shao, Y.E., Haddock, J., Runger, G., and Wallace, W.A. (1993), "Controlling the Transient Stage of a Manufacturing Process," *Proceeding of the 2nd Industrial Engineering Research Conference*, pp. 739-743.

[4] Wadsworth, H.M., Stephens, K.S., and Godfrey, A.B. (1986), *Modern Methods For Quality Control And Improvement*. John Wiley & Sons Inc., New York, NY.

Piecewise Proportional Hazards Survival Trees with Time-dependent Covariates

X. Huang, S. Chen, and S-J. Soong
Biostatistics Unit, Comprehensive Cancer Center, 153 WTI
University of Alabama at Birmingham
Birmingham, AL 35294-3300

Abstract

A tree-based method for censored survival data with time-dependent covariates is proposed. A likelihood estimation procedure is used in the recursive partitioning algorithm to grow trees. Time-dependent covariates are incorporated in the partitioning procedure under a piecewise proportional hazards structure. If time-dependent covariates are present, the estimated hazard at a node gives the relative risk for a group of individuals during a specific time period. Both cross-validation and bootstrap resampling techniques are implemented in tree selection procedure. The performance of the model is investigated through simulation and application on real data.

1 Introduction

The tree-based methods are originally used in the regression and classification (Breiman, Friedman, Olshen, and Store, 1984), later on the principle is adapted to censored survival data. The need of tree-based methods for survival data comes from clinical investigators who usually are interested in grouping patients with differing interpretable prognoses.

Including time-dependent covariates in the survival analysis leads to dynamic prognosis, where the estimated risk of the patient's survival may change from one time point to the next as the values of the covariates change. The investigation of time-dependent covariates in survival analysis has received considerable attention recently in both the statistical and biomedical literature (Cox and Oakes, 1984; Andersen, 1991).

LeBlanc and Crowley (1992) extended the proportional hazards regression to tree-structured relative risk estimates for censored survival data with one-step full likelihood estimation procedure. This method works well with time-independent covariates. Other tree-based methods have also been proposed for analyzing survival data. Gordon and Olshen (1985) presented a method using distance measures between Kaplan-Meier curves and their nearest continuous approximation. Davis and Anderson (1989) proposed a method based on the exponential log-likelihood structure. Segal (1988) presented a totally nonparametric application using the Tarone-Ware or Harrington-Fleming classes of two-sample rank statistics. LeBlanc and Crowley (1993) developed a recursive partitioning procedure based on maximizing the dissimilarity in the survival distributions of patients between regions of the covariate space. Existing survival trees methods are only suitable for dealing with censoring data with time-independent covariates. Few have been done for time-dependent covariates.

In this paper, based on LeBlanc and Crowley's work (1992), we propose a model which accommodates time-dependent covariates into piecewise proportional hazards survival trees for censored survival data (Huang, unpublished Ph.D. dissertation, Department of Biostatistics, University of Alabama at Birmingham, 1994). This methods splits nodes through the product space of the covariate and time, and establish measures of improvement based on piecewise proportional hazards. The estimated hazard function or the estimated proportionality at each branch node summarize the risk of a group of individuals during each specific time period. The next section briefly describes the basic ideas of the piecewise proportional hazards survival trees. Section 3 investigates the proposed method based on simulation studies. Section 4 exemplifies the method by an analysis of the UAB Localized Melanoma Data. Section 5 gives some discussion.

2 A New Survival Trees Method

Generally, tree-based methods recursively partition the covariate space into disjoint regions and the corresponding data into groups. For each split node some measure of separation in the response distribution between the two daughter nodes is calculated, Although many types of partitions could be considered, we will consider only splits on a single variable at a time, which is easily generalized to combinations of covariates. All possible splits for each of the covariates are evaluated, and the variable to be split and the split point are chosen to best separate the nodes. The same procedure is applied recursively to increase the number of nodes until each contains only a few observations. The resulting model can be represented as a binary tree. After a large tree is grown, there are rules for recombining nodes and for readjusting the size of the tree.

LeBlanc and Crowley's (1992) relative risk trees adopts the proportional hazards model which specifies the following hazard function at time t, for an individual with covariate vector $\mathbf{z}$

$$\lambda(t|\mathbf{z}(\cdot)) = \lambda_0(t)s(\mathbf{z}), \qquad (1)$$

where $s(\mathbf{z}) \geq 0$ and $\lambda_0(t)$ is the unknown baseline hazard. The first step of a full likelihood estimation procedure is used in a recursive partition algorithm to grow the tree. If the covariate vector $\mathbf{z}$ also changes with time, obviously, we can generalize (1) to a more general form, that is

$$\lambda(t|\mathbf{z}(\cdot)) = \lambda_0(t)s(\mathbf{z}(\mathbf{t})). \qquad (2)$$

The trees method we propose is to split nodes through the product space of the covariate and time based on a rule to minimize a loss function that is defined by the log likelihood of piecewise proportional hazards assumptions. If there are only time-independent covariates to be considered to associate with failure time, this method reduces to LeBlanc and Crowley's (1992) relative risk trees. However, we can always add an auxiliary time-dependent covariate to monitor the change of hazards with time. If time-dependent covariates are also involved, our new algorithm may give different piecewise proportional hazards survival estimates for different individuals. Even for the same individual in different time periods, he or she may be partitioned to different nodes.

If we consider time-dependent covariates that associate with the event time, the proposed piecewise proportional hazards trees method approximates the proportional hazards model (2) with the following hazard function

$$\lambda_i(t) = \begin{cases} \lambda_0(t)\theta_{i_1}, & 0 < t \leq t_{i_1}, \\ \lambda_0(t)\theta_{i_2}, & t_{i_1} < t \leq t_{i_2}, \\ \vdots & \vdots \\ \lambda_0(t)\theta_{i_k}, & t_{i_{k-1}} < t \leq t_{i_k}, \end{cases}$$

where $\lambda_0(t)$ is the baseline hazard function and θ_{i_1}, θ_{i_2}, ..., θ_{i_k} are positive. For mathematical convenience, some of t_{i_j} may be defined as ∞ so that $\lambda_i(t)$ may have less than k pieces.

For simplicity, we illustrate the case with only one time-dependent covariate which is assumed to be monotonically increasing in time for each individual. Suppose we choose a split point S_1 for a time-dependent covariate $Z(t)$. There are three possible relationships between $z_i(t)$ and S_1 for each individual: (1) $z_i(t) \leq S_1$, $0 < t \leq x_i$; (2) $z_i(t) \leq S_1$, $0 < t \leq t_{i,1}$ and $z_i(t) > S_1$, $t_{i,1} < t \leq x_i$; (3) $z_i(t) > S_1$, $0 < t \leq x_i$.

We start to grow our survival tree from the root node Γ_1, which consists of the whole sample based on a constant risk for all individuals. Under the piecewise proportional hazards assumption, it follows that the contributions of the left and right daughters of root node Γ_1 to the likelihood are

$$l(\theta_l) = \prod_{i\in I_1} (\lambda_0(x_i)\theta_l)^{\delta_i} \exp(-\Lambda_0(x_i)\theta_l) \prod_{i\in I_2} \exp(-\Lambda_0(t_{i,1})\theta_l),$$

where $I_1 = \{i \mid z_i(t) \leq S_1,\ 0 < t \leq x_i\}$ and $I_2 = \{i \mid z_i(t) \leq S_1,\ 0 < t \leq t_{i,1};\ z_i(x_i) > S_1\}$, and

$$l(\theta_r) = \prod_{i\in I_2} (\lambda_0(x_i)\theta_r)^{\delta_i} \exp(-(\Lambda_0(x_i) - \Lambda_0(t_{i,1}))\theta_r) \prod_{i\in I_3} (\lambda_0(x_i)\theta_r)^{\delta_i} \exp(-\Lambda_0(x_i)\theta_r),$$

where $I_3 = \{i \mid S_1 < z_i(t),\ 0 < t \leq x_i\}$. $\Lambda_0(t)$ is the baseline cumulative hazard function. Hence the likelihood function is

$$l(\theta_l, \theta_r) = l(\theta_l)l(\theta_r)$$

If we know the baseline cumulative hazard, the maximum likelihood estimates of θ_l and θ_r are

$$\widehat{\theta}_l = \frac{\sum_{i\in I_1} \delta_i}{\sum_{i\in I_1} \Lambda_0(x_i) + \sum_{i\in I_2} \Lambda_0(t_{i,1})}$$

and

$$\widehat{\theta}_r = \frac{\sum_{i\in I_2} \delta_i + \sum_{i\in I_3} \delta_i}{\sum_{i\in I_2} (\Lambda_0(x_i) - \Lambda_0(t_{i,1})) + \sum_{i\in I_3} \Lambda_0(x_i)}.$$

However, since we do not know the cumulative hazard, a natural estimator of the cumulative hazard given estimates $\widehat{\theta}_l$ and $\widehat{\theta}_r$,

$$\widehat{\Lambda}_0(t) = \sum_{i:x_i \leq t} \frac{\delta_i}{\sum_{j \in \{l,r\}} \sum_{i:x_i \geq t} \widehat{\theta}_j}$$

is used. Similar to LeBlanc and Crowley's model (1992), only the first iteration will be used in the recursive partitioning procedure to grow the tree. The Breslow estimator evaluated at $\widehat{\theta}_l = 1$ and $\widehat{\theta}_r = 1$, which is the Nelson (1969) cumulative hazard estimator, is used. $\widehat{\theta}_l$ and $\widehat{\theta}_r$ can be interpreted as the observed number of deaths divided by the expected number of deaths when $z_i(t) \leq S_1$ and when $z_i(t) > S_1$, respectively.

After a survival tree is grown based on the above procedure, an estimated cumulative hazard function is also obtained by iteration. Then, we take this estimated cumulative hazard function as the given baseline cumulative hazard to grow another tree from the root node, and pruning and tree selection will be based on this second tree.

If time-dependent covariates are included in splitting and growing a survival tree, it may no longer true that every split will create two exclusive individual groups according to the value of the covariate. When time-dependent covariates exist, the ratios of estimated hazards between nodes are used to summarize the relative risk of a group of individuals during a specific time period, and as the tree grows, the relative risk functions may have many pieces.

As with CART, a nested sequence of subtrees is defined by minimal cost-complexity pruning. The cross-validation and bootstrap resampling (Efron, 1982) are used to make "honest" estimates of the loss associated with each tree in the sequence and the final tree is selected based on these estimates.

3 Simulation Studies

In order to investigate the performance of the piecewise proportional hazards trees, simulation studies were conducted. In this section, procedures and results of the simulation experiments are reviewed. The comparison among the trees model and the Cox proportional hazards regression (Cox, 1972) with time-dependent covariates are studied. The proposed tree and the Cox model are applied to three types of random samples which are to be examined in different perspectives.

Each of the three simulations was designed from different perspectives. In the first simulation, the random samples were generated by piecewise exponential distributions with non-monotonous underlying hazards associated with a time-dependent covariate. In the second simulation, the random samples were generated by piecewise exponential distributions with monotonous underlying hazards associated with a time-independent covariate and a time-dependent covariate. In the third simulation the random samples were generated by Weibull distributions with mixed underlying hazards associated with a time-independent covariate, and an auxiliary time-dependent covariate was added for assessing nonconstant hazard functions. The reason for doing this was to examine the capability of the tree method dealing with different survival distributions. The Cox proportional hazards regression model was also applied to each of the random samples from three survival distributions for the purpose of comparison. In each simulation, one hundred random samples were generated.

The sample sizes are 400 for the first and the third simulations and 450 observations for the second simulation. The average censoring rate among three simulation studies is approximately 20%. The minimum terminal node size permitted for splitting was 20 observations. The right pruned subtree was selected in each method by minimizing the ten-fold cross-validation and the bootstrap estimates of the prediction error.

The piecewise proportional hazards trees performed well in all three survival distributions and basically recovered the changes of the hazard rates. The Cox proportional hazards regression model totally failed in two of our three simulations and seriously underestimated in one simulation. In the tree selection, the cross-validation procedure tended to underestimate the prediction error.

In the simulation, the proposed trees method with the add-on of auxiliary time-dependent covariates showed some strength. Our example demonstrated that with an auxiliary time-dependent covariate the proposed trees method was capable of detecting underlying hazards, which were changing with time. As the exploratory methods, the proposed trees are superior to some previous trees methods even when time-dependent covariates are not existent.

4 Example

Survival for patients with cutaneous melanoma, a cancer of the skin, is strongly associated with a number of clinical and pathological factors. in the past two decades, extensive studies have been done and remarkable progress has been made in the identification of dominant factors that affect the outcome of melanoma (Belch, Houghton, Sober, Milton, and Soong, 1992). Using the multivariate regression analysis methods for survival data, tumor thickness at diagnosis, tumor ulceration, invasion level, and lesion location are found to be the key prognostic factors for localized melanoma. Additionally, a fine model has also been developed recently to predict survival and recurrence in localized melanoma (Soong, Shaw, Balch, McCarthy, Urist, and Lee, 1992).

In this section, we reanalyze the University of Alabama at Birmingham (UAB) localized melanoma data using the piecewise proportional hazards trees method. In cancer clinical trials, discussion has largely been restricted to the analysis of mortality data, where each patient is classified as dead or alive (censored), or to the analysis of disease-free survival data, where each patient is classified as either disease-free or not. From a different perspective, with recurrence as one of the potential prognostic factors, in this analysis we would like to see the possible dynamic impact of recurrence and other factors on survival in localized melanoma. Recently, some studies have been done by considering multistate models instead of the simple two state models for survival data (Andersen, 1988). Multistate models provide a flexible framework for the study of the effects of covariates on several transition rates and important biological insight may be gained from the analysis of such a model.

The analysis presented here is based on 702 localized melanoma patients from the Surgical Oncology Service at the University of Alabama at Birmingham from 1955 to 1980. Patients have been referred primarily from Alabama, with some coming from the surrounding states of Florida, Mississippi, Tennessee, and Georgia. Approximately 78.6% of the patients had censored survival times. Four clinical and pathological factors previously known to be associated with survival are included in the analysis as time-independent covariates. They are tumor thickness, lesion location, ulceration, and level of invasion.

Tumor thickness has values 1 to 6 which are coded for tumor less than 0.76mm thick, between 0.76mm and 1.49mm thick, between 1.50mm and 2.49mm thick, between 2.50mm and 3.99mm thick, between 4.00mm and 7.99mm, and more than 8.00mm thick, respectively. Lesion location has values 0 and I corresponding with extremity and axial. Ulceration with values 0 and 1 means no and yes. Invasion with value 0 represents level II, and 1 represents level III, IV and V. In addition, we treat recurrence as a time-dependent covariate based on multiple measures recorded at each time of recurrence.

Clinically, the severity of melanoma is defined in three stages. If recurrence occurs, it must be one of the three clinical stages. Each measure of the recurrence in the analysis depends on the clinical stages of melanoma recurrence. Therefore, we code recurrence as a step-function of the time that takes values 1, 2 and 3 corresponding to the clinical stages. The recurrence time is included in this covariate. Three follow-up melanoma recurrences are used to construct the time-dependent covariate. In other words, each patient has a maximum of three possible measures for melanoma recurrence.

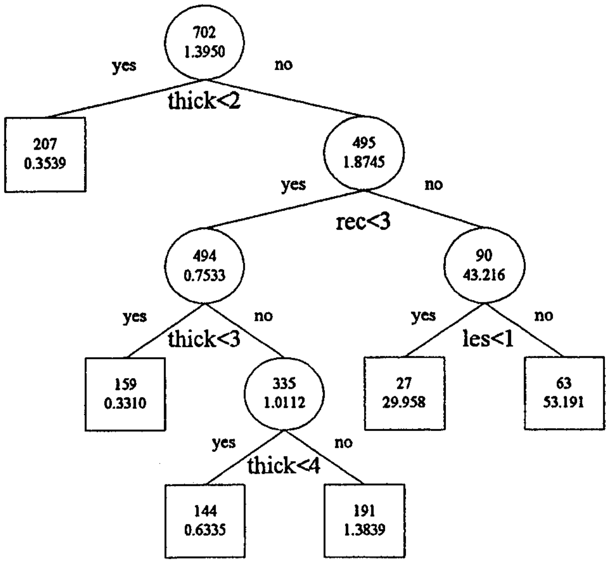


Figure 1: Survival Tree of Localized Melanoma Data

Trees were grown with a minimum node size of 25 patients. The reason for doing so is that we are not interested in extremely small prognostic groups. The piecewise proportional hazards trees grown and selected a survival tree with six terminal nodes as shown in Figure 1, where the number of patients are inside the upper level of the nodes and the estimated proportionalities are inside the lower level of the nodes. The first split to the tree was on tumor thickness with less 0.76mm thick versus thicker. The next split was on recurrence with clinical stage 3 versus other. For patients who had clinical stages 1 or 2 recurrence, the split was on tumor thickness again with between 0.76mm and 1.49mm thick versus thicker and again with 1.50mm and 2.49mm thick versus thicker. Finally, for patients who had clinical stage 3 recurrence, the split was on extremity or axial lesions.

The results were consistent with the previous analyses that tumor thickness and lesion location were important prognostic factors. Based on these brief analyses, there is evidence to show that time-dependent covariate recurrence also is a key prognostic factor. We can see clearly that patients who had the recurrence, and it changed from local to distant sites, would have much high risk of death.

Due to the adjustment of a baseline, the final piecewise proportional hazard tree shows that the tumor thickness is the most important predictor of the clinical course. Patients who had tumor less than 0.76mm thick had the best prognosis whether or not they had melanoma recurrence. In contrast, patients who had a thick tumor had a worse prognosis. However, patients who had a thick tumor could have been divided according to recurrence. Patients with stage 2 recurrence or less were doing better than patients with stage 3 recurrence, although their prognosis still could been assessed by tumor thickness in which the thicker the tumor, the worse the prognosis. Finally, patients who had stage 3 recurrence still could been partitioned with primary lesion site. Patients with axial melanomas had the worst prognosis. Again, the analysis reveals a certain interaction among these significant factors.

5 Discussion

Parallel to relative risk trees (LeBlanc and Crowley, 1992), based on a piecewise proportional hazard structure, a new survival trees methods has been proposed to appropriately handle time-dependent covariates. As more flexible alternatives to the previous works (LeBlanc and Crowley, 1992), if no time-dependent covariates are included in the data, an auxiliary time-dependent covariate could be created to monitor the change of the hazard. Even when time-dependent covariates are present, including such a covariate might fit the model better.

Simulations were conducted on each of the three data patterns with 100 repetitions, respectively. The Cox proportional hazards regression model was also applied to the random samples for comparison. The proposed trees method performed well on simulated data.

The UAB localized melanoma data set, with melanoma recurrence as a time-dependent covariate and other factors as time-independent covariates, was analyzed by the proposed trees method. The melanoma recurrence was found to be a dynamic prognostic factor that affected survival. Patients who had a recurrence that changed from local to distant sites would have little chance of surviving.

We emphasize the importance and the necessity of including time-dependent covariate in survival analysis. With time-dependent covariates, the analysis is led to dynamic prognosis. We also realize the difficulty and the complication of involving time-dependent covariates in the analysis. it is noted that time-dependent covariates, in principle, are easy to be included in the Cox model, but strict data requirements may have prevented widespread use of the Cox regression model with time-dependent covariates. However, the main problem is that the interpretation of the results from a model with time-dependent covariates is less obvious. Although we have introduced the survival trees models with time-dependent covariates, the ways we interpret the results might not be the only one or the best one. A great deal of work remains to be done.

References

Andersen, P. K. (1988). Multistate models in survival analysis: A study of nephropathy and mortality in diabetes. *Statistics in Medicine* **7**, 661-670.

Andersen, P. K. (1991). Survival analysis 1982-1991: The second decade of the proportional hazards regression model. *Statistics in Medicine* **10**, 1931-1941.

Balch, C. M., Houghton, A., Sober, A. J., Milton, G. W., and Soong, S-J. (Eds.). (1992). *Cutaneous Melanoma (2nd ed.).* Philadelphia: J. B. Lippincott Company.

Breiman, L., Friedman, J. H., Olshen, R. A., and Store. C. J.(1984). *Classification and regression trees.* Belmont. California:Wadsworth International Group.

Cox, D. R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B* **34**,187-220.

Cox, D. R., and Oakes, D. (1984). *Analysis of survival data.* London: Chapman and Hall.

Davis, P. B., and Anderson, J. R. (1989). Exponential survival trees. *Statistics in Medicine* **8**, 947-961.

Efron, B. (1982). The jackknife, the bootstrap and other resampling plans. *SIAM* **38**.

Gordon, L., and Olshen, R. A. (1985). Tree-structured survival analysis. *Cancer Treatment Reports* **69**, 1065-1069.

LeBlanc, M., and Crowley, J. (1992). Relative risk trees for censored survival data. *Biometrics* **48**, 411-425.

LeBlanc, M., and Crowley, J. (1993). Survival trees by goodness of split. *Journal of the American Statistical Association* **88**, 457-467.

Nelson, W. (1969). On estimating the distribution of random vector when only the coordinate is observable. *Technometrics* **12**, 923-924.

Segal, M. R. (1988). Regression trees for censored data. *Biometrics* **44**, 35-47.

Soong, S-J., Shaw, H. M., Balch, C. M., McCarthy, W. H., Urist, M. M., and Lee, J. Y. (1992). Predicting survival and recurrence in localized melanoma: A multivariate approach. *World Journal of Surgery* **16**, 191-195.

A Tree-based Method of Analysis for Prospective Studies

Heping Zhang, Theodore Holford, and Michael Bracken
Department of Epidemiology and Public Health
Yale University School of Medicine
New Haven, CT 06520

Abstract

Prospective studies often involve rare events as study outcomes, and it is of primary concern to identify risk factors and risk groups associated with the outcomes. Practical solutions to risk factor analyses in prospective studies are discussed. We address strategies to determine tree structures, to estimate relative risks, and to manage missing data in connection with some important epidemiological problems. Some of the basic ideas behind our strategies follow from work of Breiman, Friedman, Olshen, and Stone (1984) although we propose extensions to their methods in order to resolve some practical problems that arise in implementing these methods in epidemiologic studies.

1 Introduction

Rare events or diseases, such as AIDS and birth defects, are common targets in epidemiologic studies. Accompanying the study outcome, data on a number of putative risk factors and covariates are typically gathered. The goal is to identify risk factors associated with the outcome. Logistic and log-linear regressions, unified as generalized linear models (GLM), are popular statistical tools for analyzing these studies. One key element in the GLM is the link function between the log-odds of the events and a linear form of covariates and parameters; see McCullagh and Nelder (1989) for an excellent discussion on the subject. The GLM is very attractive in applications for many reasons such as the simplicity of the linear models and the interpretability of the parameters in the logistic models. In this paper, we present an alternative nonparametric approach which is more appropriate and flexible in many instances because this approach does not rely on most of the restrictive assumptions made in the GLM.

The tree-based method is useful to explore data when there are large numbers of variables and considerable missing information, and when a linear combination of covariates does not have an intuitive interpretation. In particular, epidemiologic studies often have categorical variables that do not have meaningful linear combinations. The tree-based method indentifies risk factors by specifying groups at risk. In the following discussion, a familiarity with the work of Breiman *et al.* (1984) is assumed.

2 The Tree-based Method

2.1 Determination of Tree Structures

Statistical inference is typically based on optimality criteria such as maximum likelihood. In the context of the tree-based methods, we use an impurity function. An advantage of this is that the statistical outputs are "best" in some specified sense. The disadvantage is that the results may not be intuitive and convenient to interpret. We now explain situations for which adjustments may need to be made for tree structures. Therefore, a *repairing* step may be required for the tree-based method. The repairing may be done during the tree growing step and/or after the tree is pruned.

In a tree the same covariate may be used to split more than one nodes while the cut-off points are different but close. We would naturally question whether the cut-off points are indeed different. This can be answered statistically using significance tests or clinically according to whether one really cares about that difference. If the answer is negative, one may want to force the cut-off to be the same so that the interpretation is simpler. Now suppose that a split on x_i is suggested by the optimality criterion in constructing a tree, but x_j is very competitive in terms of the impurity of the resulting left and right nodes. When there are other good reasons (e.g., the reliability and nature of the measurement) to use x_j, the user may prefer x_j to x_i. Moreover, if we use, say, cigarettes smoked as a covariate, suppose that the computer splits the population according to whether one smoked at least 19 cigarettes per day. It may make more sense to use 20 instead of 19 because 20 corresponds to a pack of cigarettes.

It has been observed (cf. Breiman *et al.*, pp. 313-317) that the splitting rule used in the tree growing step tends to favor end-cut splits. To avoid the end-cut preference problem, a solution provided by Breiman *et al.* is to take a different splitting criterion. A much less technical solution could be replacing the suspicious split with a competitive one that does not suffer from this problem. This is possible in our implementation of the tree-based method by allowing the user the option of selecting their own splits.

2.2 Tree Pruning

Breiman *et al.* (1984) described an automated pruning procedure via cross validation. An implicit assumption for this automated procedure is that the grown tree is intacted. As discussed in the previous section, the tree produced by an automated procedure may not be satisfactory and certain repairs may be needed. This would violate the premise under which the cross validation is used. To address this concern, we change the pruning step for the present study as follows. Since we are aiming at finding high risk individuals in a population, we will prune off a node if the risks for the left and right daughters are not significantly different at a nominal significance level in terms of the relative risk. In epidemiology, a significance level of 0.05 is usually an acceptable choice. In our pruning step, repeated significance tests are actually performed. A lower significance level may be more appropriate if our purpose is to test certain hypotheses. However, we use the significance test as a tool to select splits in the same way as in linear regression one uses the significance test to select variables via stepwise procedures. After the determination of the tree structure, our main purpose is to generate hypotheses for future studies. For this reason, it is not critical to use a "perfectly" rationale choice.

Start with a level of 0.05 and prune off a pair of left and right nodes from the bottom of the big tree if we cannot reject the hypothesis that the relative risk (estimated from the resubstitution method) for the two nodes equals 1 at this significance level. This yields a primary tree, which usually has a reasonable size. Next, we examine the primary tree to see (a) which splits are superficial by estimating the relative risk using cross validation as described below; (b) which splits may be scientifically uninterpretable by reviewing the literature; (c) which splits may need more data to justify. After this examination step, we have a final tree. Furthermore, one may use this final tree to explore alternative trees. Therefore, our pruning procedure is not completely automatic and we deliberately leave room for users to apply their knowledge of the data. See Zhang and Bracken (1994) for an application of this procedure.

2.3 Risk Estimation

In the preceding section, the relative risk is used to prune an over-grown tree. The impurity function used to select splits is closely related to the relative risk. Hence, a split of low impurity tends to result in a high relative risk. This suggests that the resubstitution estimate of the relative risk may be biased upward because impurity was the selection criterion. Despite the bias, the resubstitution estimates are still useful for pruning the large tree although they are not reliable for interpreting the final tree. Because the resubstitution estimates are upward biased, the splits of a tree tend to be more statistically significant than they really are. We expect that the number of terminal nodes after deletion using the resubstitution estimates is larger than that resulting from more realistic estimates, e.g., the cross validation method as described shortly.

To correct the bias in the resubstitution estimates, we describe an alternative method using cross validation locally. It is based on the idea that a fair estimate of relative risk may be derived from another data set that has been collected under similar conditions.

Breiman *et al.* (pp. 150–155, 1984) proposed an *ad hoc* but well-designed cross validation procedure to calculate within node misclassification rates. Unfortunately, their procedure is not directly applicable for the estimation of relative risk because: (a) the procedure focuses on the node instead of the splitting variable; (b) the relative risk may be derived from the misclassification rates if we assign the unit misclassification cost that is obviously inappropriate for the present application; and, (c) the global cross validation is not applicable when repairs must be made to the grown tree.

The local cross validation method proceeds as follows. First, we randomly divide the population of interest into v sub-populations. For instance, we may take $v = 5$. Let $\mathcal{L}_i$ $(i = 1, 2, 3, 4, 5)$ denote the 5 sub-populations. First, we leave $\mathcal{L}_1$ alone and use $\cup_2^5 \mathcal{L}_i$ to select the split s_1^* based on variable x. It is conceptually important to note that the split s_1^* is searched only over the variable that has already been chosen. This restriction is enforced in particular to address the effect of a specified factor. Then, we can use s_1^* to stratify $\mathcal{L}_1$ and record the 4 entries (a, b, c, d) in a 2×2 table based on the factor level and the response for $\mathcal{L}_1$.

	event	
	yes	no
low risk group	a	b
high risk group	c	d

Next, we repeat this process by leaving out each of $\mathcal{L}_i$

($i = 2, 3, 4, 5$) in turn and using only the remaining subpopulations to select a split s_i^* again based on race. The entries based on $\mathcal{L}_i$, which will be stratified by s_i^*, will be recorded. Sum the cell entries over the five 2×2 tables produced by the 5-fold cross validation procedure, and calculate the relative risk using the combined the 2×2 table.

Suppose that we apply the root node split in a tree to an independent, ideally similar data set, then we would have a 2×2 table, called $\mathcal{T}$, with entries (a_0, b_0, c_0, d_0). Again, a_0/b_0 is the odds for the low risk group and c_0/d_0 is the odds for the high risk group. Now, every 2×2 table obtained in the cross validation is an approximation to $\mathcal{T}$ provided that the total sample size is taken into account, despite the fact that different splits may be chosen. The combination of these 2×2 tables is a way of averaging and generally provides better approximation to $\mathcal{T}$ than those individual tables. Since potentially different splits may be chosen, the combined 2×2 table does not have an intuitive interpretation, but the combination is legitimate from a statistical point of view. Here is the reason. All individual 2×2 tables are generated by the same algorithm and the variation among them is solely due to the random sampling in the cross validation. Therefore, these 2×2 tables can be viewed as i.i.d. random 4-vectors and equal to the originally selected split in distribution. Therefore, mathematical operations on these tables are well-defined. This is a key idea behind the cross validation that is also used in Breiman *et al.* as elaborated in Remark 1.

When the selected split is not spurious but real, the constitutents of the low and high risk groups determined in the cross validation should be similar although they may be different, and hence the cross validation estimate of relative risk should be close to the resubstitution estimate. In contrast, if the selected split is spurious, the split is hardly reproducible and the constitutions of the low and high risk groups from the cross validation can be very different. The resubstitution and the cross validation estimates should also be very different. In this case, neither the resubstitution nor the cross validation method may provide an accurate estimate for the relative risk. What is important is that the spurious split is identified and the precise level of the relative risk is no longer of great interest. To some extent, we must make a subjective call on the basis of the discrepancy between the two types of estimate – the same dilemma as was seen in determining a spurious split.

It is also helpful to draw a line between an *a priori* defined split, s_0, and a split, s, selected from the impurity criteria using a learning sample $\mathcal{L}$. Where s_0 and s are conceptually different even if they are actually the same. For example, before performing the tree-based analysis we might have decided to calculate the relative risk of a disease comparing black vs white. Then, s_0 is black vs white. The relative risk can be obtained directly from the data without any adjustment because there is no bias due to the split selection. After the tree-based procedure is applied to the data, a selected split s may turn out to be s_0. This time, there is a potential bias when the relative risk for s is calculated by resubstitution. We cannot treat s in the same way as s_0 even though they look identical. Instead, this s should be regarded as the same as a split s^* which may be obtained by the same algorithm from a learning sample $\mathcal{L}^*$ that is the same as $\mathcal{L}$ in distribution.

Remark 1. This cross validation procedure is in fact inspired by the analogy with Breiman *et al.* (pp.75–78). The connection can be made as follows. As pointed out earlier, we attempt to evaluate the influence of each variable selected to form the tree. They are interested in the misclassification rate of a sub-tree corresponding to a specified complexity parameter. Therefore, two procedures are similar in the sense that they require something fixed, i.e., a variable versus a complexity parameter. In the step of using cross validation, the cut-off for the fixed variable may vary in our procedure while the structure of the sub-tree corresponding to the specified complexity parameter changes, too, in that of Breiman *et al.* Therefore, the two procedures are similar in the sense that they allow something to vary during cross validation. Finally, both procedures take an average step over the results during cross validation. Breiman *et al* (p.77) acknowledge that the cross validation estimates of misclassification rates tend to be conservative in the direction of overestimating misclassification rates. In our situation, we would then expect that the crcss validation estimate of relative risk may be biased toward the null value. Therefore, it is useful to look at both the resubstitution and the cross validation estimates of relative risk because they are potentially biased in opposite directions and presumably the resubstitution estimates are more biased.

2.4 Missing Data

In most applications of the generalized linear model, users take naive approaches to deal with missing data such as deleting subjects which have missing data in any covariates. As pointed out by Breiman *et al.* (1984, Section 5.3.2), this strategy may result in a loss of a substantial portion of the data. The nature of the recursive partitioning procedure makes it possible to handle missing data in a more efficient manner.

Two notable approaches have been proposed in the literature by Breiman *et al.* (1984, Section 5.3) and Clark and Pregibon (1992). The former uses ***surrogate splits*** to mimic the best splits. When a subject has a missing value on a covariate by which the best split is defined, a surrogate split, based on another covariate, would be used to assign the subject. We call the latter "missings together" (MT) method. In other words, the cases with missing values for the splitting variable are assigned to one node.

The strategy of using the surrogate splits fits perfectly into the tree-based method and is a most thoughtful idea. The surrogate splits are implemented in CART and can be carried over without user involvement. Nevertheless, we have a practical concern with this strategy, that is, when a reader looks at a published tree, it is not clear how a case with missing information is assigned unless the authors give all (primary and secondary) surrogate splits associated with a tree. This information is in fact available in the original CART printout, but unfortunately only a limited amount of information may be published. In classification problems, we can incorporate surrogate splits into an automated procedure without worrying about what they are. In contrast, for the present application, we must know the surrogate splits in order to report and interpret them. It could be tedious to describe all possibilities of applying the primary and secondary splits.

The MT strategy provides an alternative, simple approach for handling missing data although it may not use the data as efficiently as the surrogate splits. Now, we describe the implementation of the MT strategy. Suppose that x_i is a nominal covariate taking two distinct levels a and b (the idea extends immediately for more levels). The candidate splits accommodating missing values are NA—ab, NAa—b NAb—a, where NA stands for missing values. The idea is to treat NA as an extra level of x_i. If x_i is ordinal, Clark and Pregibon suggested the use of the same strategy by quantifying x first and then treating it as if it is nominal. Suppose that $x_i = (1,2,3,4,5,NA)'$ is an ordinal predictor, x may first be converted to, say, $\tilde{x}_i = (a,a,a,b,b,NA)'$ in which a covers 1,2,3 and b covers for 4 and 5. Then, $\tilde{x}$ would replace x in partitioning.

Clark and Pregibon's implementation of the MT strategy has two limitations. First, the quantification ignores the original order of x_i. From the example above, the natural order in $\{1,2,3\}$ versus $\{4,5\}$ vanishes. However, the order of x_i is very important for interpreting the results. Second, the quantification often uses artificially coarser measurements of the covariates than the original values of the covariates which presumably results in coarser splits. For instance, 2 in the example above can never be a cut-off value for x_i. The round-off effect may be minor, but unnecessarily.

We propose a new implementation for the MT strategy by replacing the original x with two new variables. Let $x_i = (x_{i1}, \cdots, x_{iN})'$. Define the components of $x_i^{(1)}$ and $x_i^{(2)}$ to be the same as those of x_i when the components of x_i are not missing. For all missing components of x_i, the corresponding values of $x_i^{(1)}$ and $x_i^{(2)}$ are respectively defined as $\min_j(x_{ij}) - 1$ and $\max_j(x_{ij}) + 1$. The idea is to regard the missing value as an additional distinct value of x_i. The assigned values *per se* are not important and should be viewed as a generic labeling for missing values. What is important is that the labeling ensures that all subjects having missing data will be sent to the one (left or right) side of a split. For example, taking $N = 6$, let x_i=(2.1, -4.0, NA, 1.5, 7.3, NA), then $x_i^{(1)}$=(2.1, -4.0, -5.0, 1.5, 7.3, -5.0) and $x_i^{(2)}$=(2.1, -4.0, 8.3, 1.5, 7.3, 8.3). The two copies introduced for the ordinal variable with missing values compete independently with other covariates while, obviously, only one of them may be selected at each node. For example, if $x_i^{(1)}$ is chosen as a splitting variable, it sends all subjects whose values are missing for this variable to the left daughter node. It is worthwhile to note that both $x_i^{(1)}$ and $x_i^{(2)}$ may be used again to split lower nodes thus allowing the cases with missing values to go to either side of the node.

Remark 2. We create two copies of one variable on the basis of the variable, not individual subjects. Suppose that x_1 and x_2 are two variables. If x_1 has missing values in any of its components, two copies, $x_1^{(1)}$ and $x_1^{(2)}$, will be created. Similarly, if x_2 has missing values in any of its components, two copies, $x_2^{(1)}$ and $x_2^{(2)}$, will also be created. However, if subjects 1 and 2 have missing values in both x_1 and x_2, we do not create four copies of x_1 and x_2 for subject 1 and another four copies for subject 2. We use the same copies to cover both subjects.

3 Discussion

In this paper, we have advocated the use of a tree-based method for epidemiologic studies. This nonparametric method is particularly convenient and appropriate when the objective is to identify risk factors associated with a certain event, to discover interactions among the factors, and to find high risk subpopulations. When applying CART to epidemiologic studies, some modifications are necessary. We have designed a more user-friendly program that provides users with options to control the tree structures. Prior to using the existing CART technology, there were two fundamental decisions that users would have to make: the prior probability of the outcome and the cost of misclassification. Since our data come from a prospective study, it is reasonable to estimate the prior probability from the data. Therefore, the prior selection is not a problem for us. However, it has been observed in the literature that the final tree structure is sensitive to the choice of misclassification costs [e.g., Breiman *et al.* (pp. 175–181, 1984)]. With the low prevalence rate of the outcome in the present application, it is even more difficult and subtle to specify misclassification costs and then to justify these choices. If a user changes the tree structure for various reasons, it violates the basic condition for using the procedure of Breiman *et al.* (Section 3.4.2, 1984). When this occurs, we suggest the use of an alternative pruning procedure.

Acknowledgements

This research is in part supported by NIH grants DA05484 and HD11357 (Bracken), CA30931 (Holford), and HD30712 and RR05443 (Zhang).

References

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and Regression Trees*, California: Wadsworth, 1984.

Clark, L.A. and Pregibon D. "Tree-based models" In: Chambers and Hastie ed., *Statistical Models in S.* California: Wadsworth & Brooks/Cole, 377–419, 1992.

Friedman, J. H. "Estimating functions of mixed ordinal and categorical variables using adaptive splines." *Technical Report* **108**, Dept. of Statistics, Stanford University, (1991).

McCullagh,P. and Nelder, J.A. *Generalized Linear Models* (2nd ed.), London: Chapman and Hall, 1989.

Zhang, H.P. and Bracken, M. B. "A tree-based risk factor analysis of preterm delivery and small for gestational age birth," *Amer. J. Epidemiol.* to appear, 1994.

DISCONTINUITY ESTIMATION IN NONPARAMETRIC REGRESSION VIA ORTHOGONAL SERIES

Miroslaw Pawlak

Department of Electrical and Computer Engineering
The University of Manitoba
Winnipeg, Manitoba, Canada R3T 5V6
e-mail: Pawlak@ee.umanitoba.ca

Abstract - We study the problem of detection and size measurement of discontinuities from sampled noisy data of an univariate function. The proposed tests and estimators are of the form of linear convolution filters with characteristics employing orthogonal series expansions. In particular, the standard and conjugate Fourier series are taken into consideration . The convergence properties (consistency and rate of convergence) of the proposed estimates are established.

I. INTRODUCTION

Consider an univariate regression model

$$y_i = f(x_i) + z(x_i)\ ,\ i = 1, 2,..., n, \qquad (1)$$

where x_1, x_2, ..., x_n are fixed-design points in $[-\pi, \pi]$, say, $z(x_1), z(x_2)$, ..., $z(x_n)$ are uncorrelated random errors with zero mean and finite variance σ^2 and f is the unknown regression function.

The detection of some singular points (discontinuities, corner points, etc) in an otherwise smooth function is an important problem in a number of areas as, e.g., system theory, signal/image processing and statistics [1], [5], [7], [8]. In the area of image processing a large variety of different operators for locating of changes in image intensities have been suggested. Traditionally, the proposed techniques have been introduced ad hoc and their performance has been justified by simulation studies over selected images.

More recently, however, optimal detection filters, obtained in the process of optimizing a criterion being a combination of signal-to-noise ratio, the localization measure, and resolution (quantified by number of false responses), have been proposed [2],[3], [4], [5], [6]. The local maxima in the thresholded output of such filters have been used as an estimate of the discontinuity position. No rigorous statistical analysis of the proposed techniques has been carried out.

All the above works concern finding location of the edge, they do not, however, estimate the edge size. The latter, clearly, can be an useful component in the image reconstruction and understanding processes. The problem of measurement of the size of discontinuities in a function was first (in the image processing literature) studied in [5], [6]. In [8], [9] the kernel type nonparametric regression techniques for estimating the locations of jumps points and the corresponding sizes of jump values have been proposed.

In this paper we propose a class of linear filters which are able to localize discontinuities of a function of virtually any form, i.e., the behavior of the function in the neighborhood of the discontinuity need not be in the form of step , ramp, or a polynomial of a finite order. Thus, we can copy with a nonparametric class of discontinuous functions. The proposed techniques give consistent estimates of the discontinuity size, see [10] for a complete account of our techniques. Our approach stems from the theory of Fourier series, and specifically, from results that are related to the so called Gibbs phenomenon [11]. The problem of measuring of the discontinuity size was first addressed as early as 1913 by Fejer[12], see also [11]. There, it was elaborated in the context of convergence of the partial sum of Fourier series in the neighborhood of a discontinuity of the function being expanded. Here we utilize this approach in the case when sampled noisy data generated by the regression model (1) are available. We observe also that the proposed techniques are of the form of linear filters with odd impulse response functions having multiple zeros. We prove that the filters responses at a given point converge to the discontinuity size at this point. This reveals, how the error depends on the distribution of discontinuities (our techniques allow multiple discontinuities), the function smoothness away from discontinuity, noise characteristics, sampling rate, and the filter bandwidth. As a result, an optimal value of the filter bandwidth is obtained. The problem of the discontinuity localization is also examined.

II. DISCONTINUITY MEASUREMENT AND LOCALIZATION

Let f(x) be a real, integrable function defined, without loss of generality, over $[-\pi,\pi]$. Let $\Delta(x) = f(x+) - f(x-)$ be a discontinuity size of f at the point x.

Our aim is to estimate $\Delta(x)$ using the convolution operators of the following form

$$\int_{-\pi}^{\pi} f(t)\, K_q(x-t)\, dt \quad , \tag{2}$$

where the filter characteristic $K_q(x)$ of order q satisfies two properties: (1) it is an odd function, (2) it has 2q+1 zeros in $[-\pi,\pi]$, including 0 and $\pm\pi$. Furthermore, the operator should have the property that $\int_{-\pi}^{\pi} f(t)\, K_q(x-t)\, dt \to \Delta(x)$ as $q\to\infty$ for possible general class of discontinues functions.

Filters of this form, in the context of edge detection, has been studied in the computer vision literature [1], [2], [3], [4], [5], [6]. Typically, however, the value q = 0, i.e. , only one zero-crossing at x=0, has been assumed and they not estimate the size of the discontinuity.

In this paper we propose (other alternatives are also possible) the following two prescriptions for $K_q(x)$

$$\widehat{K}_q(x) = -\frac{1}{q}\sum_{j=1}^{q} j \sin jx \quad , \tag{3}$$

$$\widetilde{K}_q(x) = -\frac{1}{\ln q}\sum_{j=1}^{q} \sin jx \quad . \tag{4}$$

Both techniques have been originated in the theory of Fourier series, see [10], [11], [12]. The kernel in (3) results from a simple integration by parts and observation that $\widehat{K}_q(x) = -\frac{1}{q}\frac{d}{dx} D_q(x)$, where $D_q(x)$ is the Dirchlet kernel of order q. The kernel $\widetilde{K}_q(x)$ is related to the theory of conjugate Fourier series [11], i.e., $\widetilde{K}_q(x) = -\frac{1}{\ln q}\widetilde{D}_q(x)$, where $\widetilde{D}_q(x)$ is the conjugate Dirchlet kernel of order q.

Since only the discrete and noisy data (1) are available one has to replace the integral in (2) by some its discrete approximation. Combining this with the definition of $\widehat{K}_q(x)$ and $\widetilde{K}_q(x)$ we can define the following estimates of $\Delta(x)$.

$$\widehat{\Delta}(x) = \sum_{j=1}^{n} y_j(x_{j+1} - x_j)\widehat{K}_q(x- x_j) \ , \tag{5}$$

$$\widetilde{\Delta}(x) = \sum_{j=1}^{n} y_j(x_{j+1} - x_j)\widetilde{K}_q(x- x_j) \ . \tag{6}$$

Our results model the performance of discontinuity estimates on grids which become increasingly fine, i.e., as $\delta_n \to 0$, where $\delta_n = \max_j (x_{j+1} - x_j)$. As a measure of discrepancy between the estimates $\widehat{\Delta}(x)$, $\widetilde{\Delta}(x)$ and $\Delta(x)$ we choose the mean square error $E\big(\widehat{\Delta}(x) - \Delta(x)\big)^2$. The behavior of both estimates is described in the following theorem.

Theorem 1. Let f be a function of bounded variation .

Then

$$E\big(\widehat{\Delta}(x) - \Delta(x)\big)^2 \approx \pi\,\sigma^2\,\delta_n q \;+\; V^2(f)\,\big(q\,\delta_n\big)^2$$

$$+ \Big(\int_{-\pi}^{\pi} f(t)\,\widehat{K}_q(x\text{-}t)\,dt - \Delta(x)\,\Big)^2 \ , \tag{7}$$

$$E(\tilde{\Delta}(x) - \Delta(x))^2 \approx \pi\sigma^2 \frac{\delta_n q}{\ln^2 q} + V^2(f)(q\delta_n / \ln q)^2$$

$$+ \left(\int_{-\pi}^{\pi} f(t)\, \tilde{K}_q(x-t)\, dt - \Delta(x) \right)^2 , \qquad (8)$$

where V(f) is a total variation of f.

The first terms in (7) and (8) are caused by the presence of noise, while the second ones represent the bias due to discreteness of data . The third terms are due a finite q used in the definition of the filter characteristics, see (3) and (4). Theorem 1 exhibits that var $\tilde{\Delta}(x)$ is smaller than var $\hat{\Delta}(x)$. As for the behavior of the last term in (7) we can show, see [10] for details, that if f(x) has right-hand and left-hand derivatives for all x ∈ $(-\pi, \pi)$ then it is of order $O(1/q^2)$. Regarding the filter $\tilde{K}_q$ we show that the last term in (8) is of order $O(1/\ln^2 q)$ assuming that f(x) is of bounded variation. Thus, the filter $\tilde{K}_q$ requires weaker assumptions than $\hat{K}_q$ in order to extract the discontinuity size. On the other hand, $\hat{\Delta}(x)$ has much smaller bias than $\tilde{\Delta}(x)$.

It is apparent that the first two terms in (7) and (8) are increasing as q becomes larger. This manifests a trade-off between random (quantified by the variance) and systematic (bias) errors. That is, to eliminate a systematic error one should use a large value of q, whereas a small value of q will reduce the random variation and discretization error.

It is evident from (7) and (8) that in order to reduce the error one has to relate q with δ_n . Hence, let $q = c\, \delta_n^{-\alpha}$, $c, \alpha > 0$. Clearly, if $\alpha > 1$ then the error tends to infinity , while for $0 < \alpha < 1$ the error goes to zero as $\delta_n \to 0$. Direct minimization of (7) and (8) with respect to q implies that the optimal q is of order $\delta_n^{-1/3}$ and $\delta_n^{-1}/\ln \delta_n^{-1}$, respectively. Furthermore, for $\delta_n \approx c/n$, $c > 0$ the corresponding errors are $n^{-2/3}$ and $1/\ln^2 n$. Hence, the estimate $\hat{\Delta}(x)$ tends to $\Delta(x)$ much faster than $\tilde{\Delta}(x)$. It is worth noting that the kernel estimate proposed in [8], [9] can reach the rate $O(n^{-2/3})$ provided that $f(x) = v(x) + \Delta\, 1_{[\theta, 1]}(x)$, where v(x) is Lipschitz continuous.

Although $\tilde{\Delta}(x)$ is slower estimate of $\Delta(x)$ it can have a better localization properties than $\hat{\Delta}(x)$. In fact, let θ be a point where the discontinuity in f(x) takes place. Assume, without loss of generality, that there is a single discontinuity and that $\Delta(\theta) > 0$. Then, clearly, $\hat{\theta} = \arg\max_x \hat{\Delta}(x)$ and $\tilde{\theta} = \arg\max_x \tilde{\Delta}(x)$ can define estimates of θ.

It can be shown [10] that $E\left(\hat{\theta} - \theta\right)^2 = O\left(n^{-4/5}\right)$ while $E\left(\tilde{\theta} - \theta\right)^2 = O\left(n^{-4/5} \ln^{-6/5}(n)\right)$. Thus, the discontinuity localization detector $\tilde{\theta}$ based on $\tilde{\Delta}(x)$ can outperform that one which uses $\hat{\Delta}(x)$.

This is a very surprising result since $\tilde{\Delta}(x)$ tends slower to $\Delta(\theta)$ than $\hat{\Delta}(x)$.
Furthermore, one can recover θ faster than $\Delta(\theta)$.
Thus, one can conclude that the problem of edge localization is "easier" than the problem of edge measurement.
All the above considerations imply that a combination of both techniques can be an attractive alternative, i.e., apply first $\tilde{\Delta}(x)$ to detect θ and then use $\hat{\Delta}(\tilde{\theta})$ as an estimate of $\Delta(\theta)$.

To illustrate the aforementioned results let us consider a piecewise constant function

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ \Delta_1 & \text{if } 0 \le x < 1 \\ \Delta_1 + \Delta_2 & \text{if } 1 \le x \end{cases}$$

Figure 1 shows $\hat{\Delta}(x)$ and $\tilde{\Delta}(x)$ for q = 10 and two different combinations of Δ_1 and Δ_2 . Figure 2, on the other hand, plots $\hat{\Delta}(x)$ and $\tilde{\Delta}(x)$ locally in the neighborhood of x=1, here q=6. The noise variance is 0.01 and n = 128. It is clear that the $\hat{\Delta}(x)$ method reaches maximum at the wrong location, i.e., x=1.15, whereas the $\tilde{\Delta}(x)$ estimate perfectly localizes the discontinuity at x = 1. Nevertheless, the value $\tilde{\Delta}(1)$ is much greater than the size of the discontinuity $\Delta(1) = 1$.

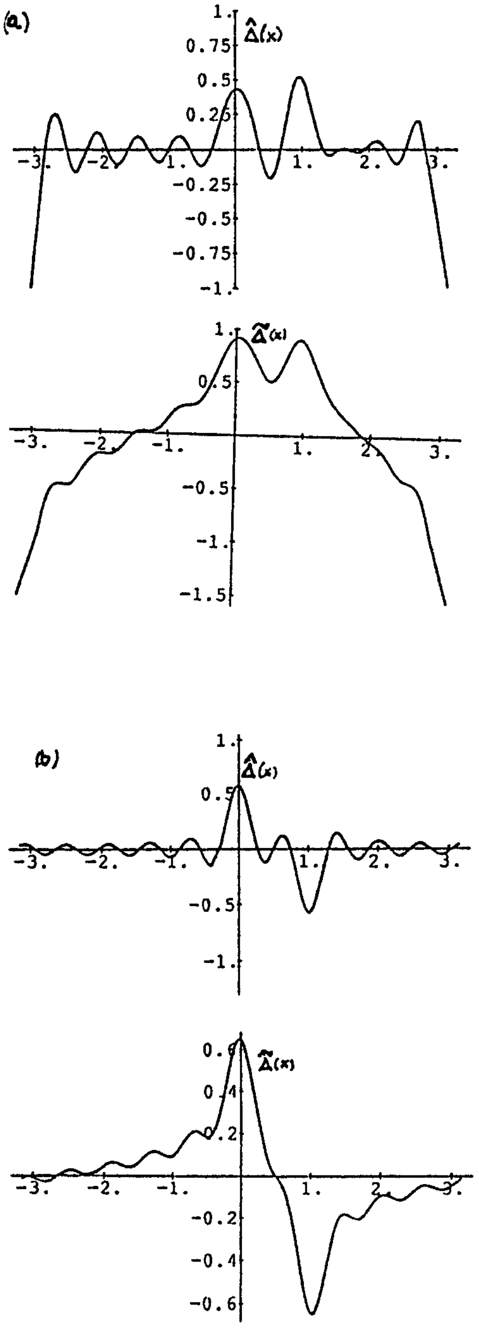


Figure 1. $\hat{\Delta}(x)$ and $\tilde{\Delta}(x)$, q= 10,
(a) $\Delta_1 = \Delta_2 = 0.5$, (b) $\Delta_1 = 0.5$, $\Delta_2 = - 0.5$

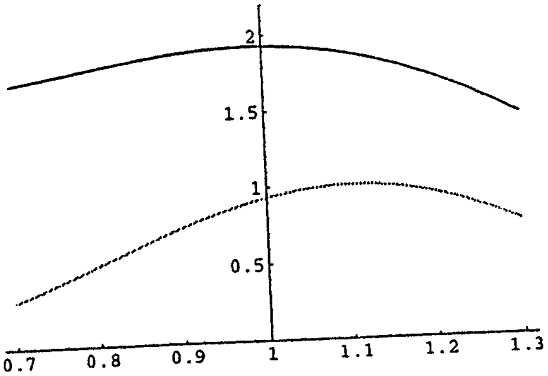


Figure 2. $\hat{\Delta}(x)$ (in gray) and $\tilde{\Delta}(x)$ (in black) in the neighborhood of x= 1; $\Delta_1 = \Delta_2 = 1$

REFERENCES

[1] V. Torre and T. Poggio, "On edge detection", IEEE Trans. Pattern Anal. Machine Intell., **8**, 147-162, 1986.

[2] J.Canny, " A computational approach to edge detection", IEEE Trans. Pattern Anal. Machine Intell., **8**, 679-698, 1986.

[3] M.Petrou and J.Kittler, "Optimal detectors for ramp edges", IEEE Trans. Pattern Anal. Machine Intell., **13**, 483-491, 1991.

[4] P.Perona and J.Malik, " Detecting and localizing edges composed of steps, peaks and roofs", in Proc. 3rd Int. Conf. Comput.Vision, 52-57, 1990.

[5] D. Lee, "Coping with discontinuities in computer vision : their detection, classification, and measurement ", IEEE Trans. Pattern Anal. Machine Intell., **12**, 321-345, 1990

[6] D. Lee, "Discontinuity detection, classification and measurment ", SIAM J. of Scientific Computing, **12**, 311-341, 1991.

[7] M.Basseville, "Detecting changes in signals and systems- a survey", Automatica, **24**, 309-326, 1988.

[8] H.G.Muller,"Change-points in nonparametric regression analysis", Ann.Stat., **20**, 737-761, 1992.

[9] J.S.Wu and C.K.Chu, "Kernel-type estimators of jump points and values of a regression function", Ann.Stat., **21**, 1545-1566, 1993.

[10] M. Pawlak, " On estimation of discontinuities in nonparametric regression", Tech. Report., 1993.

[11] A.Zygmund, **Trigonometric Series**, vols.1,2, Cambridge University Press, Cambridge, 1959.

[12] L.Fejer,"Über die Bestimmung des Sprunges der Funktion aus ihrer Fourierreihe", J.Reine Angewandte Mathematik, **142**, 165-188, 1913.

Fitting Curves with Features: Semiparametric Change-Point Methods

Paul L. Speckman *
Department of Statistics
University of Missouri-Columbia

Abstract

Change-point models have attracted attention in a variety of fields, and there are many approaches to inference, both parametric and nonparametric. This paper discusses some asymptotic results for change-point inference in the context of nonparametric regression. In one dimension, a change-point can be defined as a point with a discontinuity in one or more derivatives of the response function. A method for fitting change-points with semiparametric models is discussed which can be used with arbitrary linear smoothers. Techniques are given to identify the number and location of change-points, to estimate the size of the jump discontinuities, and to fit the entire response function with discontinuities.

KEYWORDS: change-point, nonparametric regression, semiparametric model

1. Introduction

This article reports on progress in adapting fairly general nonparametric regression smoothers to estimating curves with features such as jumps and cusps at known or unknown locations. The key idea is to use parametric models for the features (e.g. jumps or cusps) and to correspondingly modify an otherwise smooth fit to incorporate these features.

There is a very large literature on statistical methods for "change-point" problems (e.g. see Siegmund, 1986). A prototype for such problems is that of detecting a possible shift in a normal mean in independent observations over time. Assuming independent observations $y_1, \ldots, y_n$, a change occurs at time τ, $1 < \tau < n$, if

$$y_i \sim \begin{cases} N(\mu_1, \sigma^2), & i \leq \tau, \\ N(\mu_2, \sigma^2), & i > \tau, \end{cases} \tag{1.1}$$

where $\mu_1 \neq \mu_2$. The classical problems are to (i) determine if a change has occurred and (ii) if so, estimate when this happened.

A natural generalization is to relax the assumption of piecewise constant mean and to consider models in which the mean varies smoothly except for one or more isolated change-points. As an example, consider the regression model

$$y_i = \mu(t_i) + \varepsilon_i, \quad i = 1, \ldots, n,$$

where the ε_i are i.i.d. errors with $E(\varepsilon_i) = 0$, $E(\varepsilon_i^2) = \sigma^2 < \infty$. For simplicity, we will take $t_i = i/n$. A nonparametric regression version of the change-point problem is

$$\mu(t) = \begin{cases} f(t_i), & t_i \leq \tau, \\ g(t_i), & t_i > \tau, \end{cases}$$

for some $\tau \in [0,1]$, where f is a smooth function on $[0, \tau]$, g is a smooth function on $[\tau, 1]$, and $f(\tau) \neq g(\tau)$. As in (1.1), τ will be called a *"change-point"* in this setting as well. The situation where μ is continuous but has a jump discontinuity in the first derivative at some point τ can be described similarly.

There is a growing body of literature on the nonparametric regression version of the change-point problem. A parametric version in the spirit of the problem here was given by McDonald and Owen (1986), and Hall and Titterington (1992) proposed methods specific to estimating curves with peaks and edges. Müller (1992) and Wu and Chu (1993), among others, have proposed methods based on differences of nonparametric kernel estimates. Loader (1993) has recently treated change-point problems using local polynomial estimators. There is also a vast related literature in image processing devoted to edge detection (see e.g. Tagare and deFigueiredo, 1990).

The methods discussed here are applications of a general class of semiparametric models. In principle, they solve a variety of problems, and they have the advantage of not requiring specialized smoothers. Instead, the methods modify arbitrary smoothers to allow estimation and preservation of features like jumps and cusps.

2. Semiparametric change-point models

Semiparametric models for this setting go back at least as far as Wahba (1984) and Engel, Granger, Rice and Weiss (1986) for spline smoothing. Eubank and Speckman (1994) and Clive, Eubank and Speckman (1993) have recently extended these models for arbitrary linear

*Research supported in part by NSF Grant DMS-9308444

smoothers. Suppose

$$\mu(t) = \beta\phi(t_i) + f(t_i),$$

where

$$\phi(t) = \phi_k(t-\tau)$$

with

$$\phi_k(t) = \begin{cases} t^{k-1}/(k-1)!, & t > \tau \\ 0, & t \leq \tau, \end{cases}$$

for $k \geq 1$. Here f is assumed smoother than ϕ_k (i.e. $f^{(k-1)}$ is continuous at τ), and hence the parameter β is identified as $\mu^{(k-1)}(\tau+) - \mu^{(k-1)}(\tau-)$, the size of a jump discontinuity in $\mu^{(k-1)}$ at τ.

This simple model generalizes immediately to models with discontinuities in more than one derivative at τ, e.g.

$$\mu(t) = \beta_1\phi_1(t-\tau) + \beta_2\phi_2(t-\tau) + f(t),$$

or to models with multiple change-points $(\tau_1, \ldots, \tau_r)$ such as

$$\mu(t) = \sum_{j=1}^{r}\sum_{k=1}^{s_j} \beta_{jk}\phi_k(t-\tau_j) + f(t).$$

Note that the latter model has $p = s_1 + \cdots + s_r$ parameters.

We will adopt vector notation for the nonparametric regression model letting

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mu = \begin{bmatrix} \mu(t_1) \\ \mu(t_2) \\ \vdots \\ \mu(t_n) \end{bmatrix} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

and write $y = \mu + \varepsilon$. Suppose that a linear smoother is used to estimate μ. We will denote the result of smoothing by

$$\hat{\mu} = (\hat{\mu}(t_1), \ldots, \hat{\mu}(t_n))' = Sy$$

for a suitable $n \times n$ matrix S.

Assuming known change-points, a semiparametric model with p parameters can be written in terms of an $n \times p$ matrix, for example

$$X = \begin{bmatrix} \vdots & & \vdots \\ \phi_1(t_i - \tau_1) & \cdots & \phi_{s_r}(t_i - \tau_r) \\ \vdots & & \vdots \end{bmatrix}_{n\times p}$$

with $\beta = (\beta_1, \ldots, \beta_p)'$, to obtain

$$y = f + X\beta + \varepsilon.$$

2.1. Estimation with known change-points

A general method (independently derived by Denby (1986), Speckman (1988), and Robinson (1988)) for estimating β with good properties in this setting is to minimize

$$\min_{\beta} \quad \|(I-S)(y - X\beta)\|^2.$$

The solution can be expressed as $\hat{\beta} = (\tilde{X}'\tilde{X})^{-1}\tilde{X}'(I-S)y$, where $\tilde{X} = (I-S)X$.

To motivate this estimator, note that

$$(I-S)y = (I-S)f + (I-S)X\beta + (I-S)\varepsilon.$$

If f is a smooth function and S is a smoother matrix suited to the smoothness class of f, then $(I-S)f$ is negligible in comparison with $(I-S)X$, so the regression of $(I-S)y$ on $(I-S)X$ produces an approximately unbiased estimate of β. Letting $\hat{f} = S(y - X\hat{\beta})$, the entire function can be estimated as

$$\hat{\mu} = \hat{f} + X\hat{\beta} = Sy + (I-S)X\hat{\beta}. \qquad (2.1)$$

Eubank and Speckman (1991) showed that

$$\frac{1}{n}E\|\mu - \hat{\mu}\|^2 \leq \frac{1}{n}\|(I-S)f\|^2 + \frac{\sigma^2}{n}trS'S + \frac{p\sigma^2}{n}.$$

Since the first two terms on the right give the average mean square error for estimating f with smoother S, if S is any nonparametric estimator, the convergence rate is slower that $O(1/n)$, so the last term is asymptotically negligible. Thus $\hat{\mu}$ has the same global convergence properties as $\hat{f} = Sy$ does when μ has the usual smoothness assumptions. This result is independent of choice of smoother.

2.2. Example: penny data

The first example concerns the penny data given in Scott (1992) and displayed in Figure 1. The data set consists of measurements in *mils* of the thickness of a sample of 90 U.S. Lincoln pennies, two per year, from 1945 through 1989. Penny thickness was reduced in World War II, restored to its original thickness sometime around 1960, and reduced again in the '70s. Superimposed on the plot in Fig. 1(a) is a kernel smooth with bandwidth $h = 7$. Fig. 1(b) shows the fit from (2.1) with change-points for the years 1958 and 1974. With $\tau_1 = 58.5$ and $\tau_2 = 74.5$, the model

$$\mu(t) = f(t) + \beta_1\phi_1(t - 58.5) + \beta_2\phi_1(t - 74.5)$$

was estimated with a Gasser-Müller smoother. (Details on the choice of τ_1 and τ_2 are given below.)

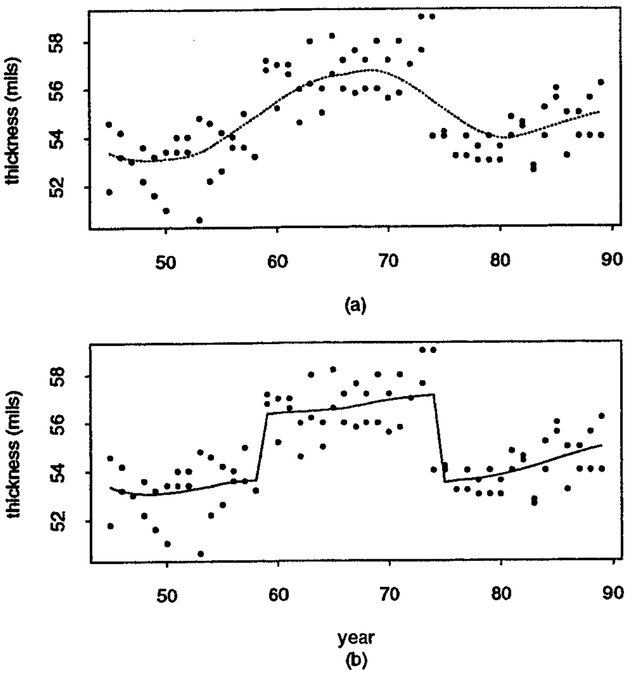


Figure 1: (a) Penny thickness data with kernel smooth, bandwidth 7. (b) Fit with change-points 58.5 and 74.5.

3. Change-point detection and estimation

In practice, the location and even number of change-points may be unknown. The strategy implemented here has three steps. First, a detection scheme is used to determine how many change-points (if any) are present. Second, the exact location of these change-points is estimated. Finally, the entire function is fit with the estimated change-points using (2.1).

3.1. Detecting one or more change-points

The semiparametric model can be used to detect change-points as follows. Suppose one is searching for a sudden change in the $(k-1)$st derivative of μ. At each of a possibly large number of candidate points τ, the model

$$\mu(t) = \beta\phi_k(t-\tau) + f(t)$$

is fit. Denote the parameter estimate as $\hat{\beta}(\tau)$. Clearly, if $\mu^{(k-1)}$ is continuous at τ, then $\hat{\beta}(\tau)$ should be near 0. But if τ is a change-point for $\mu^{(k-1)}$, then $\hat{\beta}(\tau)$ is an estimate of $\mu^{(k-1)}(\tau+) - \mu^{(k-1)}(\tau-) \neq 0$. To calibrate, let

$$Z(\tau) = \frac{\hat{\beta}(\tau)}{\sqrt{Var(\hat{\beta}(\tau))}}.$$

The problem is to determine a critical value c such that $|Z(\tau)| > c$ denotes a change-point in the vicinity of τ while controlling for false signals.

The behavior of the process $Z(\tau)$, $h < \tau < 1-h$, is detailed in Speckman (1993) under the following assumptions. Assume equally spaced points in $[0,1]$ with $t_i = i/n$, $i = 1,\ldots,n$, and further assume i.i.d. errors satisfying $E(\varepsilon_i) = 0$, $E(\varepsilon_i^2) = \sigma^2 < \infty$. To be specific, the smoother S is taken to be defined by

$$\hat{\mu}(t) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{t-t_i}{h})y_i,$$

where h is the bandwidth and K is a continuous, symmetric function with compact support $[-1,1]$. We assume further that K possesses the mth order smoothing properties $\int K(u)du = 1$, $\int u^r K(u)du = 0$, $r = 1,\ldots,m-1$, and $\int u^m K(u)du \neq 0$. (For simplicity, boundary effects are ignored, so attention is restricted to $\hat{\mu}(t), h < t < 1-h$.)

In this setting, let

$$\phi_{k\tau} = (\phi_k(t_1-\tau),\ldots,\phi_k(t_n-\tau))', \qquad (3.1)$$

and for fixed τ let $(I-S)^2\phi_{k\tau} = w(\tau) = (w_1(\tau),\ldots,w_n(\tau))'$. Then

$$\begin{aligned}\hat{\beta}(\tau) &= \frac{y'(I-S)^2\phi_{k\tau}}{\phi'_{k\tau}(I-S)^2\phi_{k\tau}} \\ &= \frac{\sum_{i=1}^{n} w_i(\tau)y_i}{\phi'_{k\tau}(I-S)^2\phi_{k\tau}}.\end{aligned}$$

It follows that

$$\begin{aligned}Z(\tau) &= \frac{\hat{\beta}(\tau)}{\sqrt{Var(\hat{\beta}(\tau))}} \\ &= \frac{\sum_{i=1}^{n} w_i(\tau)y_i}{\sigma\sqrt{\sum_{i=1}^{n} w_i(\tau)^2}}.\end{aligned}$$

Speckman (1993) showed that $Z(\tau)$ can be well approximated by a convolution process with weights $w_i(\tau)$ of the form $w(t_i-\tau)$. Figure 2(a) displays actual weights for detecting a jump in the function, i.e. $k = 1$, with $n = 100$, $h = .1$ and $\tau = .5$. Several aspects of the equivalent filter function are apparent. In Speckman (1993), it is shown that $w(t)$ has support $[-2h, 2h]$ and that $w(t) = g_+(t) - g_-(t)$, where $g_+(t) = g_-(-t)$, and g_+ is a one sided kernel satisfying $\int g_+(t)dt = 1$, $\int tg_+(t)dt = 0$ with support $[0, 2h]$. Thus the semiparametric estimate of β can be viewed as the difference of two one-sided kernel estimates. This compares with the explicit estimates constructed with differences of kernel estimates in Müller (1992) and Wu and Chu (1993), for example. Figure 2(b)

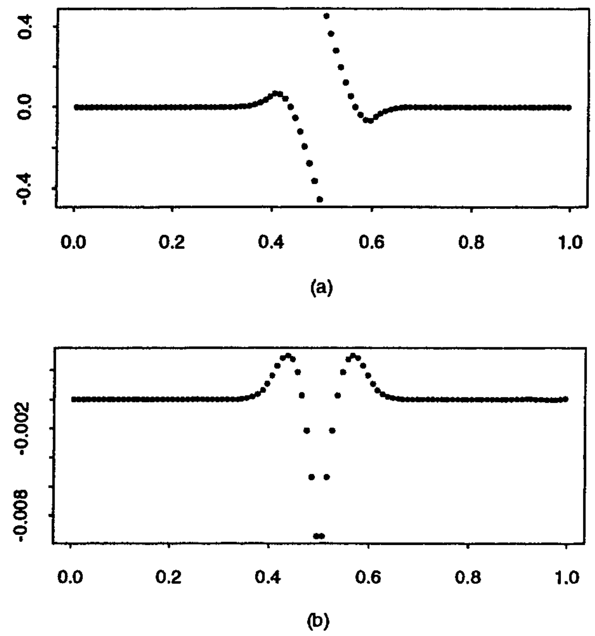


Figure 2: Actual filter weights $w_i(\tau)$ with $n = 100$, $h = .1$, (a) $k = 1$ and (b) $k = 2$.

shows weights for an example of cusp detection, $k = 2$, with $n = 100$, $h = .1$ and $\tau = .5$. Once again, $w(t)$ can be seen to to be the difference of two one-sided kernels $w(t) = g_+(t) - g_-(t)$, where now $g_+(t) = -g_-(-t)$, $\int g_+(t)dt = 0$ and $\int tg_+(t)dt = 1$.

The following theoretical results concerning the asymptotic bias of $Z(\tau)$ are obtained in Speckman (1993).

Theorem 1 *If* $\beta = \mu^{(k-1)}(\tau+) - \mu^{(k-1)}(\tau-) \neq 0$,

$$E(Z(\tau)) = \frac{\beta}{\sqrt{C_1 nh^{2k-1}}}(1 + o(1))$$

as $n \to \infty$ *for some constant* C_1 *depending only on* K, k *and* m.

If $\beta = 0$ *and* $f \in C^{2m-k}[0,1]$, *then*

$$E(Z(\tau)) \approx C_2\sqrt{n}y^{2m-k+1}f^{(2m-k)}(\tau)$$

for some constant C_2 *depending only on* K, k *and* m.

This result gives some guideline to choice of bandwidth for the detection problem. In order to avoid false detection of change-points, one needs relatively low bias. As an example, suppose a second order smoother ($m = 2$) is used. In searching for a jump in the function, $k = 1$, and it is easy to see that the bandwidth must satisfy $h = o(n^{-1/7})$ to have asymptotically negligible bias. This is a relatively large bandwidth, especially in comparison with the usual "optimal" $h \sim n^{-1/5}$ typically recommended for smoothing in this context. Note that the constant in this case depends on $f'''(\tau)$, so it is possible for finite samples that a region of high curvature could be mistaken for a jump by this method.

If one is searching for a potential change in the first derivative (a cusp), $k = 2$ and the bandwidth must satisfy $h = o(n^{-1/5})$. This suggests that undersmoothing is necessary relative to the usual widths chosen for smoothing. From a practical standpoint, if too large a bandwidth is used, a point of sharp curvature may show up as a cusp.

3.2. Choice of critical value

For fixed k, the problem is to determine a constant c_α such that

$$P(\max_{h<\tau<1-h}|Z(\tau)| > c_\alpha) \approx \alpha$$

provided $\mu^{(k-1)}(t)$ is continuous. For finite samples, this problem is not well posed because $f^{(2m-k)}(\tau)$ could be arbitrarily large. However, asymptotic results are possible. Assume the usual sequence of problems $y_{in} = \mu(t_{in}) + \varepsilon_{in}$, $i = 1, \ldots, n$, $n = 1, 2, \ldots$. If μ is fixed and sufficiently smooth and $h_n \to 0$ at a suitable rate, it is possible to find a sequence $c_{\alpha n}$ such that

$$P(\max_{h_n<\tau<1-h_n}|Z_n(\tau)| > c_{\alpha n}) \to \alpha.$$

In Speckman (1993), the asymptotic distribution is given explicitly, and simulation studies are reported. Unfortunately, the asymptotic distribution is not very accurate for finite samples, even when bias is negligible, especially for $k = 1$. For $k \geq 2$, the tube formula (c.f. Johansen and Johnstone, 1990, or Sun and Loader, 1993) can be used for improved estimation of c_α, and the author has also had some success in applying the Poisson clumping heuristic (see Aldous, 1989). However, for $k = 1$, the author has found that often the best approximation to the critical value can be obtained by a simple application of the Bonferroni inequality, $c_\alpha = z_{\alpha/2n}$, where Z_α is the $1 - \alpha$ percentile of the standard normal distribution.

Figure 3 shows the plot of $Z(\tau)$ for the penny data with $h = 4$. Here the data occurred in pairs, so the natural independent estimate $\hat{\sigma} = .820$ on 45 degrees of freedom was used. There are 45 years in the data set, and the middle 35 were searched, so the Bonferroni bound $z_{.05/70} = 3.189$ is shown. The change-points in 1958 and 1974 are clearly visible. (Note that for $k = 1$, $Z(\tau)$ is actually a piecewise constant function in τ.)

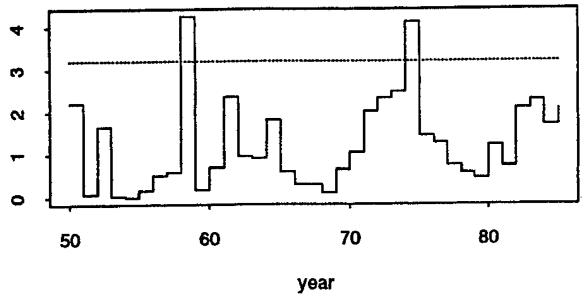


Figure 3: Plot of $Z(\tau)$ for $k = 1$, $h = 4$. Dashed line shows approximate critical value by Bonferroni for $\alpha = .05$.

3.3. Estimating change-points

Having located a change-point by the above detection scheme, the next step is to estimate the exact location. Again, this can be accomplished in principle with the semiparametric model for quite general smoothers using nonlinear least squares. For fixed k, recall the definition (3.1) of the weight vector $\phi_{k\tau}$. The (perhaps local) model

$$\mu(t) = \beta\phi_k(t - \tau_0) + f(t)$$

can be estimated by minimizing

$$||(I - S)(y - \phi_{k\tau}\beta)||^2$$

simultaneously in β and τ. Noting that

$$||(I-S)(y-\phi_{k\tau}\beta)||^2 = ||(I-S)y||^2 - \hat{\beta}(\tau)^2\phi'_{k\tau}(I-S)^2\phi_{k\tau},$$

it can be shown that $\phi'_{k\tau}(I - S)^2\phi_{k\tau}$ is essentially independent of τ, so the nonlinear least squares estimate of τ is asymptotically equivalent to the estimator which maximizes $|\hat{\beta}(\tau)|$. Thus the asymptotic distribution of $\hat{\tau}$ is obtained by studying the $Z(\tau)$ process in a neighborhood of τ_0. The asymptotics are different because β is assumed nonzero, and the situation is analogous to the results obtained by Müller (1992). Related results are obtained by Wu and Chu (1993).

The case $k = 2$ is worked out in detail in Speckman and Eubank (1994). (Similar results hold for $k > 2$.) Assume

$$\mu(t) = \beta_0\phi_2(t - \tau_0) + f(t).$$

Under the assumptions of the previous section for $m = 2$, the following results are obtained.

Theorem 2 *If $\beta_0 \neq 0$ and $hn^{1/5}$ is bounded,*

$$\sqrt{nh}(\hat{\tau} - \tau_0) \xrightarrow{\mathcal{D}} N\left(\frac{C_1\Delta_1 L}{2\beta_0}, \left(\frac{\sigma C_2}{2\beta_0}\right)^2\right),$$

where $\Delta_1 = f''(\tau_0+) - f''(\tau_0-)$, $L = \lim\sqrt{nh^5}$ and C_1 and C_2 are constants depending only on K.

Here L is defined to be zero if $h = o(n^{-1/5})$ or a nonzero constant if h converges to zero at exactly the rate $n^{-1/5}$. Note that $\hat{\tau}$ is asymptotically unbiased only if $h = o(n^{-1/5})$, i.e. if slight undersmoothing is used relative to the usual rate for best estimation of μ. Note also that if $h \sim n^{-1/5}$, $\hat{\tau} - \tau_0 = O_p(n^{-2/5})$, the best possible nonparametric rate for estimating $\mu(t)$ with two derivatives.

Asymptotic results also are available for estimating β.

Theorem 3 *Under the conditions of Theorem 2,*

$$\sqrt{nh^3}(\hat{\beta}(\hat{\tau}) - \beta_0) \xrightarrow{\mathcal{D}} N(C_4\Delta_2 L, C_5),$$

where $\Delta_2 = \frac{1}{2}(f''(\tau_0+) + f''(\tau_0-))$ and C_4 and C_5 are constants depending only on K.

Thus if $h \sim n^{-1/5}$, $\hat{\beta} - \beta_0 = O_P(n^{-1/5})$, the optimal rate for estimating $\mu'(t)$ with two continuous derivatives. Note that it is possible to estimate τ_0 better than β_0.

3.4. Data-based bandwidth choice

If the primary goal is to fit a function with discontinuities, a global estimate of μ is given by

$$\hat{\mu} = Sy + P(I - S)y,$$

where $P = \tilde{X}(\tilde{X}'\tilde{X})^{-1}\tilde{X}'$ with $\tilde{X} = (I - S)X$. Thus the influence matrix is $S + P(I - S)$, and it is natural to modify generalized cross-validation to estimate a global optimal bandwidth choice. To that end, define

$$T(y) = \frac{||y - \hat{\mu}||^2/n}{(1 - (tr(S + P(I - S))/n)^2}.$$

Another variant which might be less sensitive to undersmoothing follows a suggestion of Rice (1984):

$$T_1(h) = \frac{||y - \hat{\mu}||^2}{1 - 2(tr(S + P(I - S))/n}.$$

Note that $tr(S+P(I-S)) = tr(S)+p-tr(PS)$ since the projection matrix P has rank p by assumption. If S is symmetric with all eigenvalues between 0 and 1, it is not hard to show that $0 \leq tr(PS) \leq p$. Since $tr(S) \to \infty$ as $n \to \infty$, the terms involving P are negligible for large n. It is feasible to compute $tr(PS)$ directly by noting that

$$\begin{aligned} tr(PS) &= tr(\tilde{X}(\tilde{X}'\tilde{X})^{-1}\tilde{X}'S) \\ &= tr((\tilde{X}'\tilde{X})^{-1}\tilde{X}'S\tilde{X}). \end{aligned}$$

Since $S\tilde{X}$ can be obtained by smoothing the columns of $\tilde{X}$, the diagonal elements needed for the trace can be computed directly by matrix operations or by regressing the columns of $S\tilde{X}$ on $\tilde{X}$.

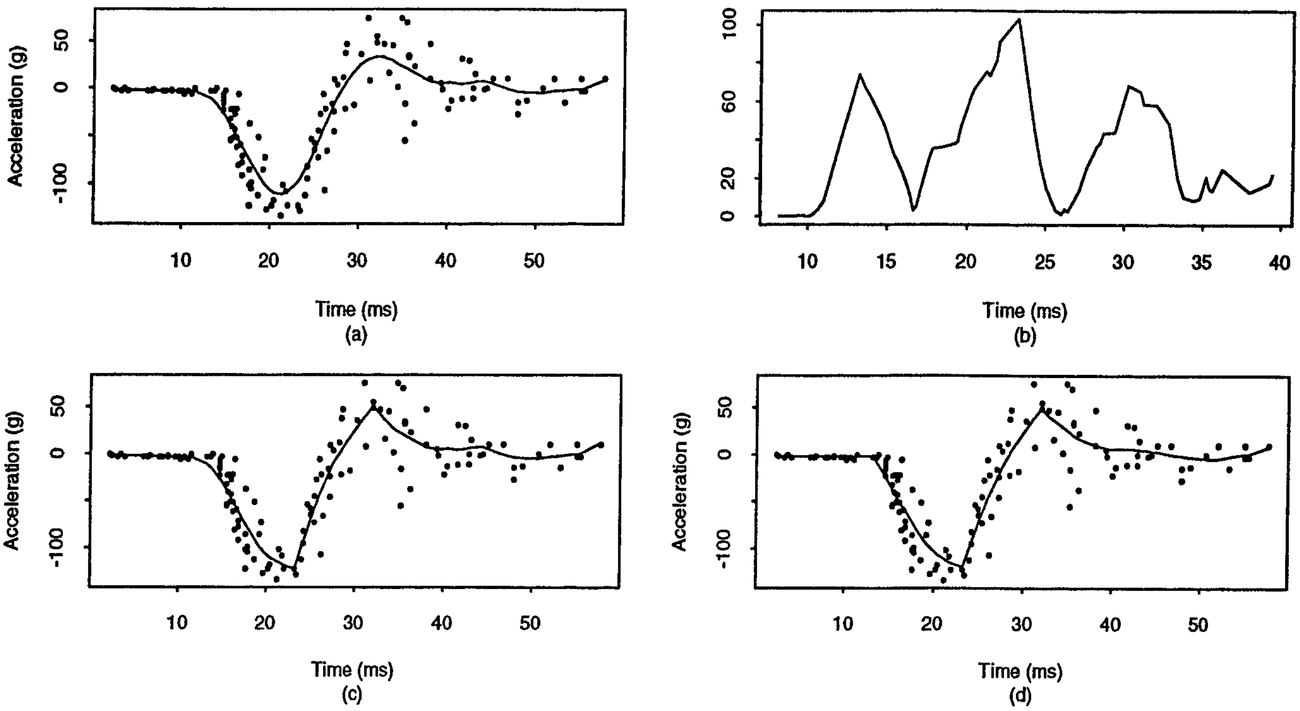


Figure 4: (a) Motorcycle data with kernel smooth, bandwidth 4; (b) $\sigma Z(t)$ versus time, $h = 7$; (c) Fit with change-points at 23.2 and 32.0; (d) Fit with initial constant to 13.2, changepoints at 23.2 and 32.0, $h = 5.2$.

4. Application to motorcycle data

To illustrate the use of change-point models for first derivatives, consider the "motorcycle data" in Silverman (1985). The data consist of 132 observations made on cadavers in simulated motorcycle collisions. The explanatory variable is time (in milliseconds) after impact, and the dependent variable is the head acceleration (in g) of a post mortem human test object. Figure 4(a) shows the data with a Gasser-Müller kernel smooth ($h = 4$ chosen by cross-validation).

The plot and the smooth show sudden changes in the direction of acceleration somewhere around 13, 23 and 32 milliseconds. In private communication, S. Portnoy has suggested that these features might be modeled as change-points in the first derivative. Such a model does not necessarily imply that there are actual corresponding physical change-points, but a model with cusps might provide a better fit to the data than smoothing and also have useful interpretation.

Figure 4(b) shows a plot of $\sigma Z(t)$ for these data. Unfortunately the data are too noisy to apply the detection criteria above, so the plot is not calibrated. For visual clarity, a bandwidth of $h = 7$ was used, and σ was not estimated. (With smaller bandwidths, the last peak is not as apparent.) There are three obvious local maxima, approximately at 13.2, 23.2 and 32.0 ms. Of course, it is very difficult to determine from the data alone if these points are "real" or if they are the result of large values of $\mu''(t)$.

The semiparametric fit with change-points 23.2 and 32.0 is shown in Figure 4(c). Unfortunately, the severe imbalance in variance between the initial data (when the head is at rest) and subsequent data prevented a good fit at the first change-point. This can be handled with weighted least squares, but an alternative strategy is presented below.

5. Constrained estimation

In the motorcycle data, it is reasonable to model position as initially constant until impact. This motivates fitting a model of the form

$$\mu(t) = \begin{cases} c, & t \leq \tau, \\ f(t), & t > \tau, \end{cases}$$

where $f(\tau) = c$ and $f(t), t > \tau$ is smooth but otherwise unspecified. (As in the treatment in the last section, this procedure can be modified to fit a function with additional cusps as well.) This problem can be addressed in several ways with semiparametric models. One method is as follows.

Since τ is a change-point of order 2, it can be located with the methodology of Section 3.3. Letting $\hat{\tau}$ denote the result, the natural estimate of c is

$$\hat{c} = \frac{1}{\#\{t_i \leq \hat{\tau}\}} \sum_{t_i \leq \hat{\tau}} y_i.$$

Thus the problem is to construct a semiparametric curve estimate, say $\tilde{f}(t)$ with a cusp at $\hat{\tau}$ satisfying $\tilde{f}(\hat{\tau}) = \hat{c}$. Then

$$\hat{\mu}(t) = \begin{cases} \hat{c}, & t \leq \hat{\tau}, \\ \tilde{f}(t), & t > \hat{\tau}, \end{cases}$$

will be an estimate with the desired properties. The required $\tilde{f}$ can be obtained by weighted least squares subject to a constraint. Consider the model $\mu(t) = \beta\phi_2(t-\tau) + f(t)$ as in Section 2., and let L' be the $n \times 1$ vector such that $L'f = f(\hat{\tau})$. Then the problem is to solve

$$\min_{\beta} \|(I-S)(y - X\beta)\|^2$$

subject to

$$L'(Sy + X\beta) = \hat{c}. \tag{5.1}$$

The solution is easily seen to be

$$\tilde{f} = \hat{f} + PL(L'PL)^{-1}(\hat{c} - L'\hat{f}),$$

where $\hat{f} = Sy + \tilde{X}\hat{\beta}$ (the unconstrained semiparametric fit) and $P = \tilde{X}(\tilde{X}'\tilde{X})^{-1}\tilde{X}'$ as before.

A mixed semiparametric model was applied to the motorcycle data. All three potential change-points were included, and the fit was subject to the constraint (5.1). Generalized cross-validation of the combined model was used to obtain a new bandwidth $h = 5.2$, and the results are displayed in Figure 4(d).

6. Summary and conclusions

Change-point problems have a long history and large literature in statistics. Ideas from change-point modeling are also very closely related to topics such as edge detection and fitting functions with features such as jumps and peaks. The semiparametric modeling discussed here provides a general and flexible way to fit models with such features using a variety of linear smoothers. These models also provide simple ways to fit functions with properties such as local constancy.

7. References

Aldous, D. (1989). *Probability approximations via the Poisson clumping heuristic.* Springer: Brooklyn, New York.

Cline, D., Eubank, R. and Speckman, P. (1993). Nonparametric estimation of regression curves with discontinuous derivatives. Unpublished manuscript.

Denby, L. (1986). Unpublished Ph.D. dissertation, University of Michigan.

Engle, R., Granger, C., Rice, J., and Weiss, A. (1986). Nonparametric estimates of the relation between weather and electricity sales. *Journal of the American Statistical Association,* 81, 310-320.

Eubank, R. and Speckman, P. (1991). A bias reduction theorem with applications in nonparametric regression. *Scandinavian Journal of Statistics,* 18, 211-222.

Eubank, R. and Speckman, P. (1994). Nonparametric estimation of functions with jump discontinuities. To appear in Proceedings of Change-Point Conference, 1992, Carlstein, Müller and Siegmund, eds.

Hall, P., and Titterington, D. M. (1992). Edge-preserving and peak-preserving estimation. *Technometrics*, 34, 429-440.

Johansen, S. and Johnstone, I. (1990). Hotelling's theorem on the volume of tubes: some illustrations in simultaneous inference and data analysis. *Annals of Statistics*, 18, 652-684.

Loader, C. (1993) Change-point estimation using nonparametric regression. Unpublished manuscript.

Müller, H.-G. (1992). Change-points in nonparametric regression analysis. *Annals of Statistics*, 20, 737-761.

McDonald, J. and Owen, A. (1986). Smoothing with split linear fits. *Technometrics*, 28, 195-208.

Rice, J. (1984). Bandwidth choice for nonparametric regression. *Ann. Statist.*, 12, 1215-1230.

Robinson, P. M. (1988) Root-N-consistent semiparametric regression. *Econometrica*, 56, 931-954.

item Scott, D. (1992). *Multivariate Density Estimation.* New York: Wiley.

Siegmund, D. (1986). Boundary crossing probabilities and statistical applications. *Anal. Statist.*, 14, 361-404.

Silverman, B. (1985). Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion). *J. Royal Statist. Soc., Series B*, 47, 1-52.

Speckman, P. (1988). Kernel smoothing in partial linear models. *J. Royal Statist. Soc., Series B*, 50, 413-436.

Sun, J. and Loader, C. (1992). Simultaneous confidence bands for linear regression and smoothing. *Ann. Statist.*, to appear.

Tagare, H. and deFigueiredo, R. (1990). On the localization performance measure and optimal edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12, 1186-1190.

Wahba, G. (1984). "Partial spline models for the semiparametric estimation of functions of several variables," in *Analyses for Time Series, Japan-US Joint Seminar*, pp. 319-329. Tokyo: Institute of Statistical Mathematics.

Wu, J. S. and Chu, C. K. (1993). Kernel type estimators of jump points and values of a regression function. *Ann. Statist.*, 21, 1545-1566.

Non-parametric Autoregressive-Regression for Edge Preserving: The Estimate and its Application in Image Processing

Hong Liu
University of Pennsylvania, Philadelphia, PA

Alexander A. Georgiev
Ethyl Corporation, Baton Rouge, LA

Abstract A nonparametric algorithm for restoring digital images corrupted with additive noise is presented. Itt is assumed the noisy image is realization of a spatial autoregressive process which also has a regression component. On the basis ofo the nonparametric functional estimation theory, a nonparametric estimate of the image is given as a restoration result. The edge preserving issue is under consideration in this study. With the proper selection of the algorithm's parameters, the estimate can preserve step edges while suppressing noise. The proposed algorithm is not sensitive to the estimation accuracy of the parameters, and can be run almost as quickly as a local averaging filter.

1. Introduction

Image restoration is a process to recover the original image from degradations due to blurring and noise corrupting. In this paper, the degradation sources are limited to additive noises. The most common model in this setting is the following

$$y_{ij} = f(i, j) + \epsilon_{i,j} \qquad (1)$$

where i, j = 0, 1, ..., m-1, $\{y_{ij}\}$ is the degraded observation, $f_{ij} = f(i,j)$ is the original signal, $\{\varepsilon_{ij}\}$ is a zero mean noise that may contain outliers. The assumption on ε_{ij} implies any restoration procedure must be resistant to the outliers, or robust restoration. In statistical point of view, image restoration under this model can be given by a robust regression. This approach, however, can have three problems. Firstly, an ordinary parametric regression procedure will make various assumptions on the signal function and the distribution of noises, which limits their practical usefulness. Secondly, model (1) does not capture the nonstationarity of the image random field and so current smoothing techniques have various limitations in performance. And finally, these smoothers lack efficient means to accommodating the need for edge preserving. Recently, nonparametric functional estimation theory[1] provides us some versatile regression tools that can be utilized to recover noise-degraded images.

It is well known that image restoration is an ill-posed inverse problem, i.e., no unique solution exists. Hence, Previous studies either employ minimum mean square error (MMSE) criteria or Bayesian analysis to estimate $\{f_{ij}\}$ given observations $\{y_{ij}\}$. In the MMSE approach, images are initially assumed to be a stationary random field, and later to be a nonstationary mean, nonstationary variance (NMNV) image stochastic model[2]. The Bayesian approach to restoration is based on the *a priori* knowledge of the statistical properties of the ensemble of objects $\{f\}$. This usually takes the form of a Markov random field[3,6]. The use of local properties is the characteristic of both NMNV and Markovian models. In this paper, local dependence is explicitly modeled as a spatial autoregressive process compounded with a regression component, which we call the AutoRegressive-REgression (ARRE) model. Parametric linear ARRE models have been studied by Ripley[4] and Cliff and Ord[5]. This paper extends the linear ARRE to be a general ARRE model which is adaptive to the image. The smoothing algorithm based on this model can smooth out both additive noises and outliers (impulse noises) while preserving sharp edges and corners and therefore keeping most details clear. It is very time efficient as well. If no impulse noise is present, restoration can be done in one loop on an image matrix. In addition, each pixel can be processed separately without waiting for the results of its neighboring pixels. This makes it suitable for parallel processing.

2. ARRE Image Model and Its Nonparametric Estimator

Let $N_{uv} = \{y_{ij} \mid \text{grid point } (i,j) \text{ is within a local neighborhood of } (u,v)\}$, where (u,v) is not necessarily a grid point. The ARRE image model assumes

$$y_{ij} = g(N_{ij} , i , j) + \epsilon_{ij} \qquad (2)$$

where i, j = 0, 1, ..., m-1, $f_{ij} = g(N_{ij},i,j)$ is the unknown image function that needs to be estimated by a restoration procedure. In contrast with this model, we would like to call Eq.(1) the REgression (RE) model. We assume that this image field $\{y_{uv}\}$ satisfies the φ-mixing condition[7]. We also assume that random field $\{y_{uv}\}$ is homogeneous in φ-mixing, i.e., coefficient φ_r does not depend on position (u,v). To give an ARRE restoration, we only need to know r_φ which determines the size of neighborhoods N_{uv}.

The nonparametric estimator for RE model (1) and their properties have been well studied by researchers[8,9,10]. ε_{ij}'s can be both dependent and non-identically distributed random variables satisfying the φ-mixing condition. One recent research which is analogous to our study in the time domain is the nonparametric prediction for an autoregressive time series, by Collomb[7] who studies the autoregressive time series in $\mathbb{R}^p$ is of the form: $y_i=r(y_{i-1}, ..., y_{i-s})+\varepsilon_i$.

To give a nonparametric estimator for the ARRE model (2), we assume observations $\{y_{ij}\}$ come from a φ-mixing random field and letting N_{ij} s contain equal number of pixels. The neighborhood structure N_{uv} is determined by the φ-mixing condition of the given image. In this study, for the simplicity, we empirically use a four-neighbor system for each pixel. N_{uv} is then the following 1x5 vector:

$$N_{uv} = \begin{cases} (y_{i-1\,j-1} , y_{i-1\,j} , y_c , y_{i\,j-1} , y_{i\,j}) & if\ (i-1<u<i)\wedge(j-1<v<j) \\ (y_{i-1\,j} , y_{i\,j-1} , y_{ij} , y_{i\,j+1} , y_{i+1\,j}) & if\ (i=u)\wedge(j=v) \end{cases}$$

where the center pixel $y_c = (y_{i-1\,j-1}+y_{i-1\,j}+y_{i\,j-1}+y_{ij})/4$ is the initial value for the interpolation. For the purpose of robustness against outliers, we let y_{ij} be the center of N_{ij} and leave this center pixel out of the summations in (3). We then measure the distance of N_{uv} and N_{ij} by the following weighted sum of squares:

$$dist(N_{uv} , N_{ij}) = w_1(N_{uv}(1)-N_{ij}(1))^2+w_1(N_{uv}(2)-N_{ij}(2))^2+w_0(N_{uv}(3)-N_{ij}(3))^2 \\ +w_1(N_{uv}(4)-N_{ij}(4))^2+w_1(N_{uv}(5)-N_{ij}(5))^2$$

where $N_{uv}(k)$ refers the k-th element of N_{uv}, w_1 and w_0 are weights satisfying $w_0 > w_1 > 0$ and $w_0 + 4w_1 = 1$. These weights play important roles in regulating smoother's outlier resisting and detail preserving abilities. Our experiment results show this four-neighborhood system works well for various types of images. We define dist(·,·) as the square of Euclidian type of distance function just for the convenience of analyzing the mean distance Edist(N_{uv},N_{ij}). We denote the restored image by $\{f_{mij}\}$, where mxm is the image size. We give a nonparametric estimate of ARRE model (2) in the following equation (3).

$$f_{muv} = \frac{\sum_{i\neq u}\sum_{j\neq v} y_{ij}k_{1,h_1}(dist(N_{uv},N_{ij}))k_{2,h_2}(u-i)k_{2,h_2}(v-j)}{\sum_{i\neq u}\sum_{j\neq v} k_{1,h_1}(dist(N_{uv},N_{ij}))k_{2,h_2}(u-i)k_{2,h_2}(v-j)} \qquad u,v=0,1,...,m-1. \qquad (3)$$

where $k_{i,h_i}(\cdot)=k_i(\cdot/h_i)$, i=1,2, are two kernel functions[1].

This estimate can be justified in two ways. Firstly, note that the random sequence in [7] is implicitly assumed to be generally nonstationary. When we explicitly include a deterministic spatial variable (i,j) to capture the nonstationarity and assume image signals are φ-mixing, the course of proof of the asymptotical properties[7] is still valid as long as the joint density function of y_{ij} and N_{ij} is continuous in the spatial position variable (i,j).

Secondly, if we let $k_{1,h_1}(\cdot)$ be the rectangular kernel, (3) will degrade to the RE estimator when window size h_1 is large enough and (u,v) locates itself within the smooth area of the image. The ARRE model is then reduced to RE model. It is worth noting again that, in [10], random noises do not necessarily have to be identically distributed nor independent of each other. What the ARRE model and its estimator differ from RE model is the results in edge areas and in the areas with discontinuities. Our restoration algorithm based on the ARRE model preserves edges and details while RE model does not. In finite sample situation, we have found through simulations that the triangular kernel based estimate performs better than the rectangular one in suppressing certain type of impulse noises but worse in smoothing regular noises. In our image restoration experiments in section 3, we always let $k_{1,h_1}(\cdot)$ be the rectangular kernel.

There are four parameters h_1, h_2, w_0, and w_1. We believe that regular MSE-based cross-validation[7] with a smoothness constraint is no longer sufficient for edge preserving and the MSE of the second order derivatives should be included in the objective function of the optimization. While this is the direction to go, the possible performance improvement of using this type of optimization would be restrained by the complexity of the restoration filter. Our approach in this paper is to consider the effects of these parameters separately in the following way.

We determine h_1 by studying the mean distance $\mathbf{\mathit{E}}\text{dist}(N_{uv},N_{ij})$. For the ease of analysis, we assume the noise field $\{\varepsilon_{ij}\}$ is a zero-mean independent sequence with a standard deviations (std) σ_n. Outliers are zero-mean with a std σ_o which is significantly greater than σ_n.

It is easy to calculate that, in the smooth area with no outliers, $\mathbf{\mathit{E}}\text{dist}(N_{uv},N_{ij}) = 2\sigma_n^2$, and in the smooth area with an outlier, $\mathbf{\mathit{E}}\text{dist}(N_{uv},N_{ij}) = w_0(\sigma_o^2-\sigma_n^2) + 2\sigma_n^2$. When f_{uv} and f_{ij} are located on the opposite sides of an edge that has a grey level contrast d, $\mathbf{min}_{\{(u,v),(i,j)\}}\mathbf{\mathit{E}}\text{dist}(N_{uv},N_{ij}) = w_0d^2+2\sigma_n^2$. When f_{uv} and f_{ij} are located on the same side of this edge, $\mathbf{max}_{\{(u,v),(i,j)\}}\mathbf{\mathit{E}}\text{dist}(N_{uv},N_{ij}) = 4w_1d^2+2\sigma_n^2$. Therefore, for the object of robust restoration, we should choose h_1 such that

$$2\sigma_n^2 < h_1 < w_0(\sigma_o^2-\sigma_n^2) + 2\sigma_n^2\ . \tag{4}$$

For the object of edge-preserving restoration, we should choose h_1 such that

$$4w_1d^2+2\sigma_n^2 < h_1 < w_0d^2+2\sigma_n^2\ , \tag{5}$$

where d is the minimum grey level contrast of all the edges that need to be preserved. An additional condition on the weights follows from (5): $4w_1 < w_0$. Obviously, to select h_1 to satisfy both (4) and (5), we have to have $\sigma_o^2 > d^24w_1/w_0 + \sigma_n^2$.

As for the value of h_2, since it is the parameter to control the balance between fidelity and smoothness, we choose its value according to the nature of the data. In this image processing application, we choose to use either 2.5 or 3.5 to let the weighted averaging (3) take place over 5×5 or 7×7 windows in the image plane.

Note that the denominator in (3) will become zero on the edges or outlying noise corruption occurs. To distinguish outliers from edge elements, a local statistical test is sufficient when it is pluged in (3). A complete nonparametric ARRE image restoration algorithm can be found in [13].

3. Experimental Results

In all the experiments, we use $w_0 = 0.6$ and $w_1 = 0.1$ so that $w_0 + 4w_1 = 1$ and $4w_1 < w_0$. We let h_1 be $(4w_1d^2+2\sigma_n^2 + w_0d^2+2\sigma_n^2)/2$, h_2 be 2.5 (5×5 averaging window) except for the tool image where h_2=3.5 (7×7 averaging window). For each experimental object, we run the ARRE restoration algorithm twice, called twicing. That means that, after we get first output from the algorithm, we take this output as the input of the second run. The results are summarized in Table 1 in terms of the Signal-to-Noise Ratio (SNR) improvement. Actual photos of all images can be found in [13].

4. Conclusions

In this paper, we present an edge-preserving restoration algorithm based on the nonparametric estimation of an autoregressive-regression model. We also demonstrate its performance by the experiments. It smooth both additive noise and additive impulse noise while preserving details including sharp corners. Whereas the priority is detail-preserving, the balance between noise suppressing and detail preserving can be adjusted by the input parameters d and h_2. Because of the nonparametric nature, no assumption is required concerning the distribution and the independence of the noises.

In this algorithm, we need one ***a prior*** information σ_n, the standard deviation of the additive noise, to determine the parameter h_1. A method that estimates σ_n directly from the degraded image can be found in [12]. We can also simply estimate σ_n in a flat area of the image. In [13], we showed that the ARRE restoration algorithm is not sensitive to the estimation error of σ_n and d.

References

[1]. Hardle, W., Applied nonparametric regression. New York: Cambridge University Press, 1989.

[2]. Kuan, D. T., Sawchuk, A. A., Strand, T. C., and Chavel, P., Adaptive noise smoothing filter for images with signal-dependent noise. IEEE Trans. Pattern

Anal. Mach. Intell., Vol. PAMI-7, No.2, 1985, 165-177.

[3]. Geman, S. and Geman, D., Stochastic relaxation, Gibbs distribution, and Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell., Vol.PAMI-6, 1984, 721-741.

[4]. Ripley, B. D., Spatial statistics, Chapter 5, 88-95, John Wiley & Sons, New York, 1981.

[5]. Cliff, A. D. and Ord, J. K., Spatial processes: models and applications, Pion Limited, London, 1981.

[6]. Woods, J. W., Two-dimensional discrete Markovian fields. IEEE Trans. Infor. Theory., IT-18, 1972, 232-240.

[7]. Gyorfi, L., Hardle, W., Sarda, P., and Vieu, P., Nonparametric curve estimation from time series. Springer-Verlag, Berlin, 1989.

[8]. Georgiev, A. A., Local properties of function fitting estimates with application to system identification. Mathematical Statistics and Applications, Proceedings, 4th Pannonian Symp. Math. Statist., Sept. 4-10, 1983, Bad Tatzmannnsdorf, Austria (W. Grossmann et al., Eds.), pp. 141-151. Reidel, Dordrecht, 1985.

[9]. Georgiev, A. A., Consistent nonparametric multiple regression: the fixed design case. Journal of Multivariate Analysis, Vol.25, 1988, 100-110.

[10]. Fan, Y., Consistent nonparametric multiple regression for dependent heterogeneous processes: the fixed design case. Journal of Multivariate Analysis, Vol.33, No.1., 1990, 72-88.

[11]. Chan, P. and Lim, J. S., One-dimensional processing for adaptive image restoration. IEEE Trans. Acoust., Speech, Signal Processing, Vol.ASSP-33, 1985, 117-126.

[12]. Meer, P., Jolion, J., and Rosenfeld, A., A fast parallel algorithm for blind estimation of noise variance. IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI-12, No.2, 1990, 216-223.

[13]. Liu, H., A Nonparametric Autoregressive-Regression Model for Robust Data Smoothing and Its Application to Image Restoration, Ph.D. dissertation, Medical University of South Carolina, 1992.

	degraded image	**ARRE restoration**	**ARRE twicing**	**recursive median**
simulated image	19.880	28.807	33.656	30.652
girl image	16.541	23.936	23.203	18.923
tool image	12.053	22.592	27.387	21.932
scene image	18.534	21.550	22.775	22.660

Table 1. Signal-to-noise ratio improvement with the ARRE restorations.

Epi Meta -- Meta-Analytic Statistical Software For Epidemiological Studies*

P. A. Hartford[1], J. R. Menkedick[1], Paul Feder[1], G.D. Williamson[2]
[1]Battelle, Statistics and Data Analysis Systems
[2]Centers for Disease Control and Prevention, Epidemiology Program Office

Abstract

Epi Meta, Version 1.2, is a meta-analysis software package developed for the Centers for Disease Control and Prevention (CDC) with the intent of distribution to those who analyze public health data. The software was designed to 1) be user-friendly from both a software and a statistical point of view; 2) provide meta-analysis options not previously available to this group of users in a menu-driven user-friendly package; 3) include the features necessary to produce a meta-analysis for particular data structures without needing additional software or knowledge of a programming language; and 4) interface with CDC's Epi Info comprehensive data management and data display system.

Throughout the design and implementation of Epi Meta a balance was maintained so that the novice user would not be overloaded with too many decisions, yet the more experienced user would not be limited to a "canned" meta-analysis. Built into the system are default choices which provide the novice user with a standard meta-analysis and enough summary information and graphs to understand program output and analysis results. Program and system design and architecture features include heavy emphasis on graphical displays to evaluate the fitted models, and menus to make it easy to choose and iterate on models and data. Although Epi Meta is primarily focussed on the meta-analysis of dose-response studies of relative risks, the underlying methods are much more widely applicable. The analysis methods fit straight-line relations within each study relating relative risk to exposure dose using transformations and weighted least squares. Goodness-of-fit of the dose-response model is assessed and an outlier analysis is performed by means of graphical and tabular diagnostic displays. The comparison across studies takes the resulting slopes and intercepts from the within study analysis and individually and jointly compares the results using fixed and random effects inferences. Epi Meta uses menus and on-screen information to guide the user through the analysis of the multiple individual studies and the comparison across all the studies. Integrated into the package is the data management facility of Epi Info, allowing for easy data entry and editing of the data throughout the meta-analysis session. The software was developed by Battelle under Contract No. 200-87-0540 with CDC. This paper discusses the design decisions involved in producing a stand-alone statistical software package through presentation of the decisions made in developing Epi Meta.

Overview

Epi Meta was developed by Battelle for CDC to provide public health officials with a tool to help them draw appropriate inferences when combining results across multiple epidemiological studies. The software is oriented toward epidemiological studies and is intended to be used by public health officials with an excellent understanding of the data but not necessarily advanced training in either statistical theory or computer programming. The statistical methods to be included in the software were determined by a literature review of methodology and application papers dealing with meta analyses of epidemiologic and medical studies and by CDC's experience with the target audience of public health officials. The literature review revealed one striking dissimilarity between the statistical approaches suggested in the methodological papers and those actually utilized in application papers. The methodology discussions nearly unanimously recommended the use of random effects model based inference procedures, whereas almost all the meta-analysis applications reviewed were based on fixed effects models and inference procedures. Therefore, a primary goal of Epi Meta was to provide user-friendly software that offered easy implementation of the random effects model to address the disparity between the methodological recommendations of the statisticians and the methodological practice carried out by the medical and epidemiological meta-analysts.

Other requirements included:

* Development of this software was carried out by Battelle under Contract No. 200-87-0540 with the Centers for Disease Control Epidemiology Program Office.

1. The package needed to be user-friendly from a computing standpoint, i.e., it would not require any computer programming and would be easy to learn to use. As a corollary, the program documentation needed to be equally user-friendly.

2. The package needed to be user-friendly from a statistical standpoint, providing:

   a. a default analysis but also alternative analyses, options and diagnostic displays and statistics.
   b. the ability to quickly and easily iterate on the analysis — deleting studies, choosing transformations, changing models, making predictions, etc.

3. The package needed to be portable and widely available, not requiring expensive or specialized software or hardware.

The software and statistical design decisions made in developing Epi Meta are illustrative of programming issues that are characteristic of the development of stand-alone statistical software for customized applications. These distinctive statistical programming requirements arise from the fact that in developing user-friendly customized statistical software for individuals who are not professional statisticians, there is not necessarily a single series of steps or a single right answer for any given analysis and certainly no right answer for all analyses. The added layer of complexity in statistical programming comes with the difficult decisions concerning such issues as:

- how much of a canned analysis (a black box) do you provide
- how many options do you make available
- how much guidance do you give
- how many warnings and diagnostic tools are required
- how do you lead the unsophisticated user to conclusions while still guarding against inappropriate inferences.

Most often, statistical analysis is dynamic and iterative, leading to the additional question of how do you develop a software interface and output so that it is easy to iterate and the user has the information necessary to perform these iterations.

The software design decisions that address these specific statistical programming needs involve the user-friendliness of the system (from a computing standpoint): use of a menu-driven system, on-line help, ease of data entry and data editing, etc. The statistical design decisions that address these needs can be viewed in a hierarchical manner. On the first or primary level are the statistical methodology and programming decisions that are made by the software developers. These decisions are embedded in the product and transparent to the user, noted only in the technical software documentation. An example of this level of decision in Epi Meta would be the weighting algorithm (Dersimonian and Laird) used in the random effects analysis. On the second level are decisions made by the developers that cannot be changed by the users but which are noted in the user's program output as informative messages. An example of this in Epi Meta is the use of the theoretical weighted residual mean square (WRMS) of 1 if there is not significant heterogeneity in the weighted linear regression for determining the within study dose-response slope. On the third and highest level are decisions that are so specific to an individual analysis or so capable of changing the results of the analysis that they are incorporated as user options in the software package. An example of this level of decision in Epi Meta is the choice of whether to use an intercept or no-intercept model in determining the within-study dose-response slope. Further examples of these types of decisions are given in the presentation of Epi Meta that follows.

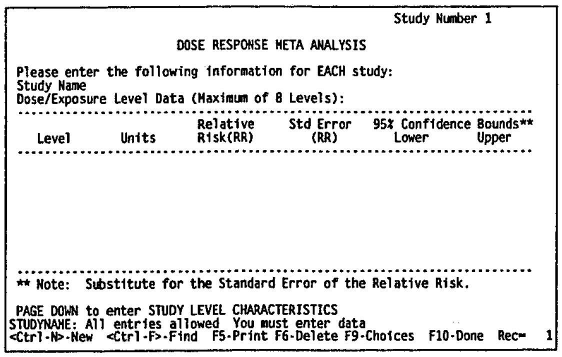
Epi Meta

Data Management System

As mentioned above, a key requirement of user-friendly statistical software is the ability to quickly and easily iterate on the analysis. This, in turn, requires ease of data entry and editing. In the case of Epi Meta, it was essential that users be able to easily add and remove dose/exposure levels and studies. For this reason, Epi Info, a CDC-distributed data management and analysis software package familiar to many public health officials, was chosen as a data management "front-end" to Epi Meta. Epi Meta transparently calls Epi Info's data management facilities to allow the user to create, edit and save data sets within an Epi Meta session.

Epi Meta allows the user to create two types of data files. The first type is for the case where there are multiple dose levels per study.

Figure 1 illustrates the first data entry screen for this type of file. For each study in the analysis, the user can enter up to eight dose levels. Within each dose level the user enters the units of the dose level, the relative risk, and the



```
                                                  Study Number 1
                       DOSE RESPONSE META ANALYSIS

 Please enter the following information for EACH study:
 Study Name
 Dose/Exposure Level Data (Maximum of 8 Levels):
 ...........................................................................
                                Relative    Std Error   95% Confidence Bounds**
    Level        Units          Risk(RR)       (RR)        Lower        Upper
 ...........................................................................









 ...........................................................................
 ** Note:  Substitute for the Standard Error of the Relative Risk.

 PAGE DOWN to enter STUDY LEVEL CHARACTERISTICS
STUDYNAME: All entries allowed  You must enter data
<Ctrl-N>-New  <Ctrl-F>-Find  F5-Print F6-Delete F9-Choices  F10-Done  Rec=   1
```

Figure 1. Data Entry in Epi Meta

standard error or upper and lower 95% confidence bounds for the relative risk.

The types of statistical decisions mentioned above are already operative at this stage of Epi Meta. For instance, if the user enters both the standard error and the 95% confidence interval, the program will use the standard error as the measure of variability for that exposure level, representing a primary level decision transparent to the user.

The second type of data file allowed is for the case where there is a single dose/exposure level per study. In this case, for each study the user enters the study name, the relative risk, and the standard error or upper and lower 95% confidence bounds for the relative risk.

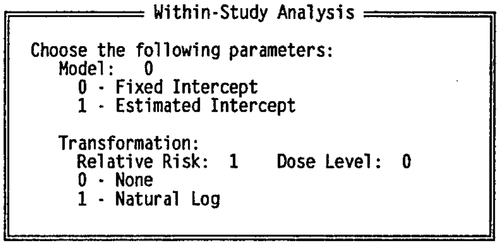
The data entered into either of these study level data files is not limited to dose/exposure levels and relative risks. An option allows for entry of user-defined response variables and an associated measure of variability. However, for this discussion, a series of dose levels with associated relative risks within each study will be used for illustration of the software.

After the data have been entered into Epi Meta, the analyst has the opportunity to edit the data prior to analysis. The analysis process is divided into two portions, a "within-study" analysis and an "among-study" analysis. Each portion is discussed in turn below.

Within-Study Analysis

The within-study analysis provides the user the ability to calculate the summary statistic(s) for each study (a slope, or a slope and an intercept) that will subsequently be used in the among-study meta-analysis. The within-study analysis is only operative in the case where there are multiple dose/exposure levels per study.

Once the data file for analysis has been specified, a within-study analysis options screen is presented as illustrated in Figure 2.

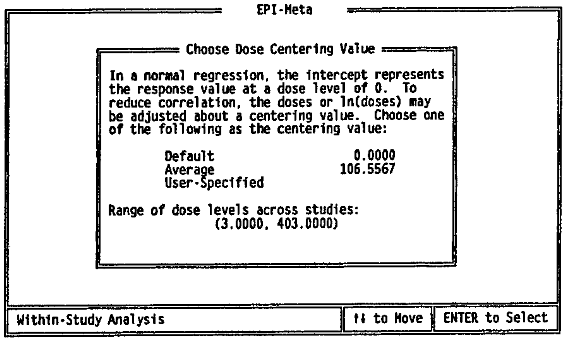


```
═══════════════ Within-Study Analysis ═══════════════

  Choose the following parameters:
     Model:   0
       0 - Fixed Intercept
       1 - Estimated Intercept

     Transformation:
       Relative Risk:  1      Dose Level:  0
       0 - None
       1 - Natural Log
```

Figure 2. Within-Study Options Screen

This screen is an example of the third and highest level of decisions required in statistical programming: those decisions that are presented as options to the user. Even here, however, there are difficult primary level choices to be made, for only a subset of all possible options will be made available to the users. The options presented allow the user reasonable flexibility in determining the type of analysis without providing all possibilities. In Epi Meta, the user has a choice of intercepts and data transformations, but is limited to the straight line model and weighted linear regression chosen by the software developers. The user's data transformation options are limited to a natural log transformation. As explained earlier, defaults are provided at all levels of the program as guidance for less experienced users. These defaults represent a compromise between providing a "black box" analysis and providing a wide variety of options. In this case, the default option for the user is to run the within-study analysis using a fixed intercept model, a log transformation of the relative risk and no transformation of the dose level. Decisions concerning the default choices are often very difficult. For example, in the case of Epi Meta, many epidemiologists prefer a fixed intercept model because of the contention that the relative risk at dose zero is known to be 1. On the other hand a statistician may prefer the use of a variable intercept model to allow for a better fit within the range of the data if there is non-linearity at low dose levels. Because the target audience for this software is public health officials, the fixed intercept model was chosen as the default.

Since Epi Meta was designed primarily as a tool for public health officials, many of the design decisions, such as the use of defaults, were made with the intention of helping those target users. However, other options were included to allow users to modify the analysis. A good example of this occurs in the case where the user chooses the variable intercept model. Here the user is given the option to center the dose levels, reducing correlation between the estimated intercept and slope for each study. The screen for this option is illustrated in Figure 3.



```
EPI-Meta

            Choose Dose Centering Value
   In a normal regression, the intercept represents
   the response value at a dose level of 0.  To
   reduce correlation, the doses or ln(doses) may
   be adjusted about a centering value.  Choose one
   of the following as the centering value:

           Default                     0.0000
           Average                   106.5567
           User-Specified

   Range of dose levels across studies:
                  (3.0000, 403.0000)

Within-Study Analysis                  ↑↓ to Move    ENTER to Select
```

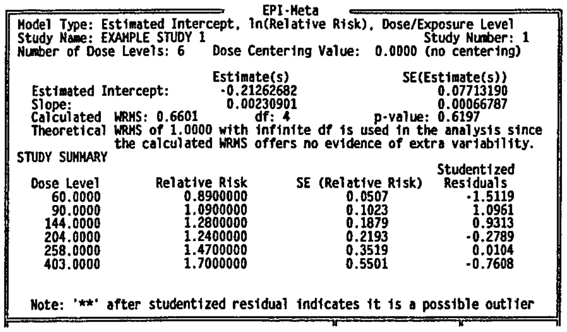
Figure 3. Dose-Centering Options Screen

The average dose level is provided for those that would like to center the doses at the mean level, and the range of dose levels is provided for those who wish to specify some other level. This on-screen information is an example of a second level design decision that is user-friendly both from a computing and a statistical standpoint. Note that if the user specifies a dose level outside the range of the reported dose levels, the user is warned and asked for confirmation. Here the user is provided guidance, but left with the final analysis option. The default option is no dose centering, allowing users unfamiliar with the reasons for choice of a centering value to pass through this stage.

The actual calculation of the within-study slope and intercept involves many primary and secondary level decisions, including the appropriate weighting scheme for the weighted linear regression, the heterogeneity test, and an algorithm for determining the appropriate weighted residual mean square. In general, options for primary and secondary level decisions are not offered to the users for one of two reasons: 1) the chosen methodology is determined to be most appropriate; or 2) the different options that could be made available would have only a minor effect on the calculated results.

Presentation of diagnostics and results in Epi Meta includes both numerical and graphical displays and output. Many primary and secondary level decisions are made here to help determine both the manner in which unsophisticated users are led to appropriate conclusions and the amount of information available to more sophisticated users to evaluate and choose among results.

An example of the first screen of numerical output in Epi Meta for the within-study analysis is presented in Figure 4.



```
                              EPI-Meta
Model Type: Estimated Intercept, ln(Relative Risk), Dose/Exposure Level
Study Name: EXAMPLE STUDY 1                              Study Number: 1
Number of Dose Levels: 6    Dose Centering Value:  0.0000 (no centering)

                              Estimate(s)              SE(Estimate(s))
  Estimated Intercept:        -0.21262682                   0.07713190
  Slope:                       0.00230901                   0.00066787
  Calculated  WRMS: 0.6601          df: 4          p-value: 0.6197
  Theoretical WRMS of 1.0000 with infinite df is used in the analysis since
              the calculated WRMS offers no evidence of extra variability.
STUDY SUMMARY
                                                           Studentized
  Dose Level         Relative Risk        SE (Relative Risk)  Residuals
     60.0000             0.8900000               0.0507          -1.5119
     90.0000             1.0900000               0.1023           1.0961
    144.0000             1.2800000               0.1879           0.9313
    204.0000             1.2400000               0.2193          -0.2789
    258.0000             1.4700000               0.3519           0.0104
    403.0000             1.7000000               0.5501          -0.7608


  Note: '**' after studentized residual indicates it is a possible outlier
```

Figure 4. Within-Study Output

The first few lines of the output are devoted to summarizing the user-specified input decisions made earlier. Next, a summary of the within study analysis, the parameter estimates and the associated standard errors, is listed. Immediately following the estimates is the goodness-of-fit result discussed above. The calculated WRMS and degrees of freedom are listed as well as the associated p-value for the chi-square statistic which is used as a test of heterogeneity. If the p-value is not significant at the 5 percent level, a note is placed on the next line letting the user know that an internal decision has been made to use the theoretical WRMS of 1 with infinite degrees of freedom. This message alerts the user to a statistical methodology decision that may not have been anticipated, while still maintaining internal control of the analysis.

The last half of the page provides a summary of the study, listing for each dose level, the response variable, the standard error of the response, and the studentized residual. An outlier test is performed on the studentized residuals, flagging possible outliers with a double asterisk. Here the analyst can use either the double asterisk as an indicator of a possible outlier or examine the actual studentized residuals.

Graphical diagnostics are provided through three types of graphs: 1) a normal probability plot of the studentized residuals, 2) a plot of the studentized residuals versus the dose levels, and 3) a plot of the estimated line and the observed relative risks, appropriately transformed. These graphs were chosen because they help the user visually assess in a simple and straightforward manner 1) the normality of the data; 2) possible outliers; and 3) the fit of the model to the data. An example of the type of diagnostic graphical display available is illustrated in Figure 5.

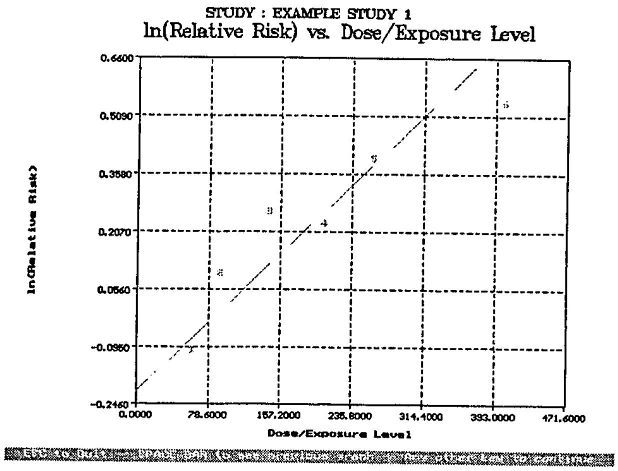


Figure 5. Within-Study Diagnostic Graph

Among-Study Analysis

All the features and statistical analysis of Epi Meta discussed so far were designed to provide the input information and capabilities required to conduct the among-study meta-analysis. The input for the among-study analysis is either the file created by Epi Meta in the Within-Study Analysis or the Single Dose Level Per Study data file created using the data management system, where both files contain the point estimate(s) and their associated standard error(s) for each study which will be used in the meta-analysis.

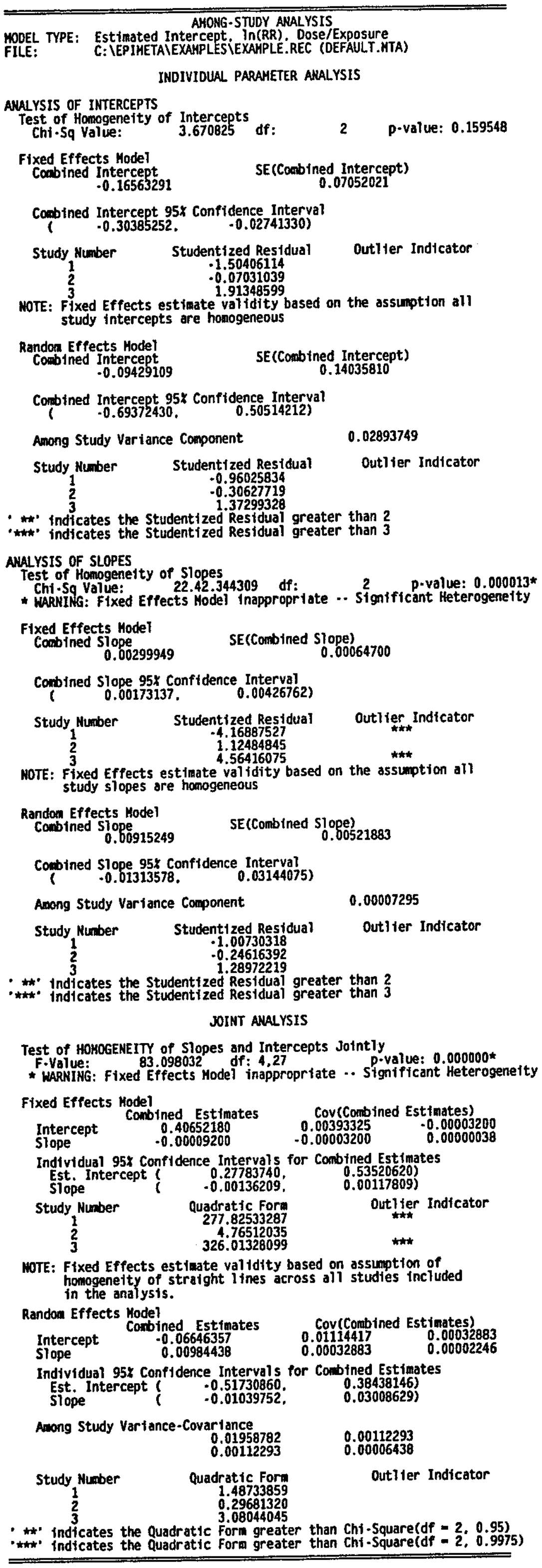
A slightly different approach was taken for handling the statistical programming decisions in the among-study meta-analysis as compared to the within-study analysis. The user is given no control over the type of analysis to be conducted. The only program choice made available to the user is the choice of the data file on which to run the analysis. Rather than allow the user to choose certain analysis options, such as a fixed versus random effects model, or an individual versus joint parameter analysis, the decision was made to present all results for several selected analyses, along with certain warnings and guidance. Two levels of output reports are offered: a "Complete Analysis Report" and a "Summary Only" report. Therefore, in a similar fashion to the within-study analysis, the user is given options concerning which analysis to use; however, the results for all options are always presented. In a like manner, a "default" analysis is available in the form of the summary report, where the summary results presented are offered as the default "answer".

As in the within-study analysis, the most difficult statistical design decisions occurred in determining which results should be presented, in what manner, and with what degree of interpretation and guidance. Again both numerical and graphical displays and output were provided.

An example of the "Complete Analysis Report", which presents all analysis results and graphs for individual and joint parameter analysis, is provided in Figure 6.

The first three lines of the "Complete Analysis Report" provide general information concerning the model on which the individual study parameter estimates are based and the file used for the analysis. This kind of simple user-friendliness from a computing standpoint also has benefits from a statistical standpoint, making for easier iteration and comparison of analysis. The first set of results presented is the individual parameter analysis (slope or slope and intercept). The F-value or chi-square statistic, degrees of freedom, and the associated p-value for the test of homogeneity of the parameter across studies is listed. If significant heterogeneity is indicated, an asterisk (*) is placed after the p-value and a warning is placed on the line immediately after the test of homogeneity letting the user know that the fixed effects model estimates are judged inappropriate because of significant heterogeneity. The flag and warning strike a compromise between refusing to present the results because they are judged questionable by the software developer's judgement, and presenting the results with no guidance or warning when inferences based on the analysis might be inappropriate.

Immediately following the test of homogeneity of the parameter is the fixed effects model analysis. The combined parameter estimate and standard error of this estimate are listed along with 95% confidence intervals. An outlier analysis is presented, listing both the studentized residual and an outlier indicator, similar to that found in the within study analysis. The final line of the fixed effects model analysis always reminds the user of the assumptions on which the model is based, i.e. the validity of the fixed effects model estimates is based on the assumption that all study parameters are homogeneous. The output for the random effects model for the individual parameter estimates is similarly displayed if the among-study variance component estimate is positive. If this estimate is not positive, then only the among-study variance component estimate is listed along with a warning that the random effects analysis is not estimable.



```
                              AMONG-STUDY ANALYSIS
MODEL TYPE:  Estimated Intercept, ln(RR), Dose/Exposure
FILE:        C:\EPIMETA\EXAMPLES\EXAMPLE.REC (DEFAULT.MTA)

                          INDIVIDUAL PARAMETER ANALYSIS

ANALYSIS OF INTERCEPTS
  Test of Homogeneity of Intercepts
    Chi-Sq Value:        3.670825   df:         2     p-value: 0.159548

  Fixed Effects Model
    Combined Intercept                SE(Combined Intercept)
             -0.16563291                        0.07052021

    Combined Intercept 95% Confidence Interval
      (      -0.30385252,         -0.02741330)

    Study Number          Studentized Residual      Outlier Indicator
         1                      -1.50406114
         2                      -0.07031039
         3                       1.91348599
  NOTE: Fixed Effects estimate validity based on the assumption all
        study intercepts are homogeneous

  Random Effects Model
    Combined Intercept                SE(Combined Intercept)
             -0.09429109                        0.14035810

    Combined Intercept 95% Confidence Interval
      (      -0.69372430,          0.50514212)

    Among Study Variance Component                      0.02893749

    Study Number          Studentized Residual       Outlier Indicator
         1                      -0.96025834
         2                      -0.30627719
         3                       1.37299328
' **' indicates the Studentized Residual greater than 2
'***' indicates the Studentized Residual greater than 3

ANALYSIS OF SLOPES
  Test of Homogeneity of Slopes
    Chi-Sq Value:       22.42.344309   df:        2     p-value: 0.000013*
  * WARNING: Fixed Effects Model inappropriate -- Significant Heterogeneity

  Fixed Effects Model
    Combined Slope                SE(Combined Slope)
             0.00299949                        0.00064700

    Combined Slope 95% Confidence Interval
      (      0.00173137,         0.00426762)

    Study Number          Studentized Residual      Outlier Indicator
         1                      -4.16887527               ***
         2                       1.12484845
         3                       4.56416075               ***
  NOTE: Fixed Effects estimate validity based on the assumption all
        study slopes are homogeneous

  Random Effects Model
    Combined Slope                SE(Combined Slope)
             0.00915249                        0.00521883

    Combined Slope 95% Confidence Interval
      (      -0.01313578,         0.03144075)

    Among Study Variance Component                      0.00007295

    Study Number          Studentized Residual      Outlier Indicator
         1                      -1.00730318
         2                      -0.24616392
         3                       1.28972219
' **' indicates the Studentized Residual greater than 2
'***' indicates the Studentized Residual greater than 3

                              JOINT ANALYSIS

  Test of HOMOGENEITY of Slopes and Intercepts Jointly
    F-Value:        83.098032    df: 4,27           p-value: 0.000000*
   * WARNING: Fixed Effects Model inappropriate -- Significant Heterogeneity

  Fixed Effects Model
                    Combined  Estimates        Cov(Combined Estimates)
    Intercept           0.40652180          0.00393325       -0.00003200
    Slope              -0.00009200         -0.00003200        0.00000038
    Individual 95% Confidence Intervals for Combined Estimates
      Est. Intercept (        0.27783740,          0.53520620)
      Slope          (       -0.00136209,          0.00117809)
    Study Number              Quadratic Form              Outlier Indicator
         1                      277.82533287                   ***
         2                        4.76512035
         3                      326.01328099                   ***
  NOTE: Fixed Effects estimate validity based on assumption of
        homogeneity of straight lines across all studies included
        in the analysis.
  Random Effects Model
                    Combined  Estimates        Cov(Combined Estimates)
    Intercept          -0.06646357          0.01114417        0.00032883
    Slope               0.00984438          0.00032883        0.00002246
    Individual 95% Confidence Intervals for Combined Estimates
      Est. Intercept (       -0.51730860,          0.38438146)
      Slope          (       -0.01039752,          0.03008629)

    Among Study Variance-Covariance
                                 0.01958782          0.00112293
                                 0.00112293          0.00006438

    Study Number              Quadratic Form              Outlier Indicator
         1                        1.48733859
         2                        0.29681320
         3                        3.08044045
' **' indicates the Quadratic Form greater than Chi-Square(df = 2, 0.95)
'***' indicates the Quadratic Form greater than Chi-Square(df = 2, 0.9975)
```

Figure 6. Among-Study Output

The results of the joint parameter analysis are printed immediately following the individual parameter analysis. Included in this analysis is an F-value or chi-square statistic for a joint test of homogeneity of the slope and intercept along with the associated degrees of freedom and p-value. Similar to the individual analysis, if the p-value is less than 0.05 an asterisk is printed after the p-value and a warning printed immediately following the results cautioning the user that the fixed effects joint estimates may be inappropriate.

The fixed effects combined weighted joint estimate of the intercept and slope, the variance-covariance matrix of the joint estimates, the 95% confidence limits for both the intercept and slope, and the joint studentized residuals, flagged if the study is a possible outlier, are all listed.

The random effects model results for the joint analysis are presented next if, analogous to the random effects individual parameter analysis, the among-study variance-covariance matrix estimate does not have all elements equal to zero. If the among-study variance-covariance matrix estimate does have all elements equal to zero, then the random effects estimates are not presented and an appropriate warning is listed. Otherwise, output similar to the fixed effects estimates joint parameter analysis is listed.

The last part of the output, not shown in Figure 6, presents a summary of the data used in the meta-analysis. This allows the user to store the actual data with the results for future reference.

Note that Figure 6 shows warnings in two places that the fixed effects model is inappropriate. The results obtained from the fixed effects individual parameter analyses are inconsistent with those from the fixed effects joint parameter analyses. This inconsistency does not occur for the random effects analyses. Thus the unsuspecting user is warned of impending pitfalls.

As shown, the "Complete Analysis Report" presents the user with both fixed and random effects model based inferences for both an individual and joint parameter analysis, with certain flags, warnings and guidance. Many primary level decisions concerning the statistical methodology (where different reasonable options were possible) made during the among-study analysis are explained to the user only in the technical appendix to the user documentation. These include, for the individual parameter estimates: the estimate of degrees of freedom for the fixed effects standard error, the joint analysis test of heterogeneity, the degrees of freedom for the test of "significant" studentized residuals, the method of estimating the random effects variance component, and the estimate of the degrees of freedom for the random effects standard error. For the joint parameter estimates they include: the degrees of freedom for the fixed effects variance-covariance matrix, the estimate of the among-study variance-covariance matrix, the degrees of freedom associated with the among-study variance-covariance matrix, the estimate of joint residuals, and the estimate of joint confidence levels on predicted values. As mentioned for the within-study analysis, these decisions that are functionally transparent to the user are usually imbedded in the software because either 1) the chosen methodology is determined to be most appropriate; or 2) the different options that could be made available would have a minor effect on the calculated results. In Epi Meta, decisions concerning appropriate methodology were based on a prior literature review of methodology and applications papers and represent the current recommended methodology. However, there are some cases where evolving methodology must be incorporated into a statistical program. One methodology choice in Epi Meta that is transparent to most users who do not read the technical documentation, is the ad-hoc method of estimating the random effects among-study variance-covariance matrix in the joint parameter analysis. As discussed in the technical documentation, the among-study variance-covariance matrix is not invariant to the choice of a value on which to center the dose/exposure levels. Therefore alternative choices of a centering constant may result in differing values of the joint analysis among-study variance-covariance matrix. In general, the more customized the statistical application, and the more advanced the methodology, the more difficult will be the choices concerning which decisions should and should not be embedded in the software.

The graphical displays and diagnostics associated with the complete analysis report include, for each model and each parameter, the following graphs: 1) a normal probability plot of studentized residuals; 2) a plot of studentized residuals versus the study number; 3) a plot, for each parameter (slope and intercept), of the estimated parameter versus the study number; 4) a plot of the quadratic forms of the joint residuals versus the study number for each model; and 5) a plot, for each individual parameter analysis, of the parameter estimates for each study with associated 95% confidence bounds versus the overall estimate and its 95% confidence limits. An example of the diagnostic plot of the quadratic forms of the joint residuals versus study number is provided in Figure 7 below.

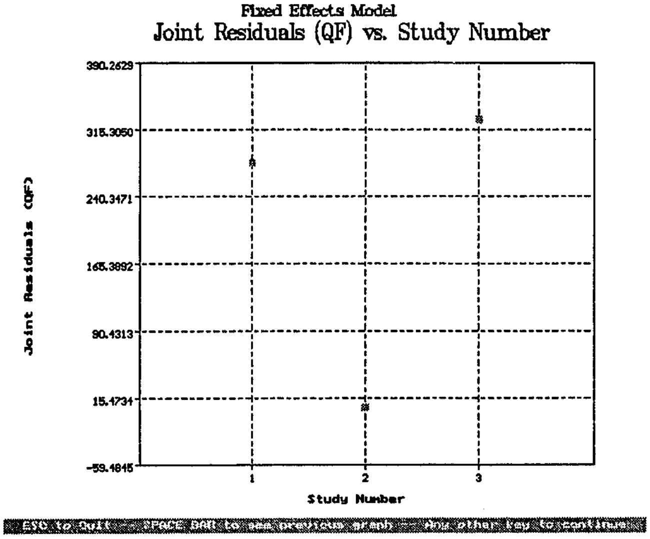

Fixed Effects Model
Joint Residuals (QF) vs. Study Number
Joint Residuals (QF)
390.2629
315.3050
240.3471
165.3892
90.4313
15.4734
-59.4845
1
2
3
Study Number
ESC to Quit -- SPACE BAR to see previous graph -- Any other key to continue

Figure 7. Among-Study Diagnostic Graph

An example of the plot of the overall meta-analysis estimate and associated confidence intervals along with the individual study results is provided in Figure 8.

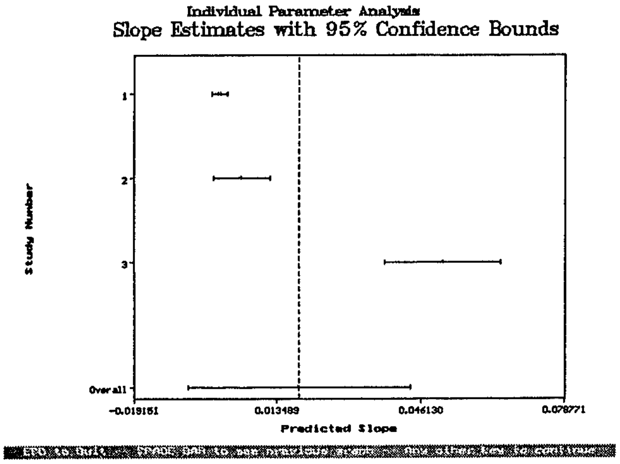
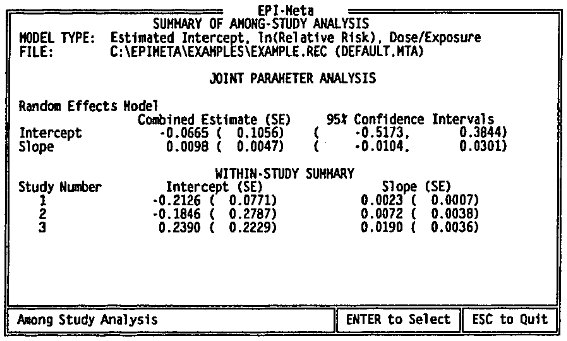

Individual Parameter Analysis
Slope Estimates with 95% Confidence Bounds
Study Number
1
2
3
Overall
-0.019151
0.013489
0.046130
0.078771
Predicted Slope
ESC to Quit -- SPACE BAR to see previous graph -- Any other key to continue

Figure 8. Among-Study Summary Graph

The "Summary Only" report provides a compressed summary of the results of the analysis providing a "default" best estimate of the combined weighted slope or combined weighted slope and intercept. This report is for the user who does not wish to choose between the alternative analyses presented in the Complete Analysis Report. The decision on which analysis to present is based on a decision tree determined by the software developers. If the fixed intercept model was used to combine the data within each study, then the random effects model estimates are given provided the variance component is positive, otherwise the fixed effects model estimates are given. When the estimated intercept model is used to combine the information within each study and the among-study variance-covariance matrix does not have all elements equal to zero, the joint analysis random effects model estimates are presented, otherwise the joint analysis fixed effects model estimates are given. An example of the Summary Report output is presented in Figure 9 below.



```
                                EPI-Meta
                   SUMMARY OF AMONG-STUDY ANALYSIS
MODEL TYPE:  Estimated Intercept, ln(Relative Risk), Dose/Exposure
FILE:        C:\EPIMETA\EXAMPLES\EXAMPLE.REC (DEFAULT.MTA)

                         JOINT PARAMETER ANALYSIS

Random Effects Model
                 Combined Estimate (SE)     95% Confidence Intervals
Intercept           -0.0665 (  0.1056)    (      -0.5173,        0.3844)
Slope                0.0098 (  0.0047)    (      -0.0104,        0.0301)

                          WITHIN-STUDY SUMMARY
Study Number         Intercept (SE)                  Slope (SE)
    1               -0.2126 (  0.0771)             0.0023 (  0.0007)
    2               -0.1846 (  0.2787)             0.0072 (  0.0038)
    3                0.2390 (  0.2229)             0.0190 (  0.0036)
```

Figure 9. "Summary Only" Report

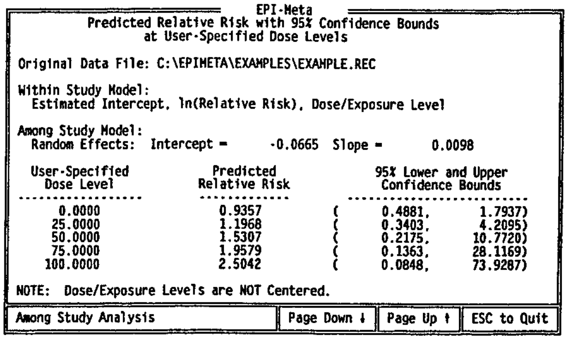
Similar to the Complete Analysis Report the first three lines of the numerical Summary Report provide a summary of the model used to combine the data within each study and the file from which the data came. Next, the default best estimates and the standard errors of the estimates are presented in accordance with the above described rules. Finally, a brief summary of the actual data used in the meta-analysis is given.

Only one type of graph is provided in the Summary Report. For each study, the parameter estimates (slope and intercept) with the associated 95% confidence bounds are displayed in comparison with the overall default best model parameter estimate and associated 95% confidence interval (See Figure 8 above).

A final among-study analysis output option allows the interested user to generate predicted relative risks and associated 95% confidence bounds for various dose levels. The model used to generate these estimates is determined using the same decision tree discussed above. The user can enter up to five dose levels and is warned if any of the dose levels are outside the range of the model.

Figure 10 illustrates the output provided when user-specified predictions have been chosen.

The first half of the output page summarizes both the within and among-study analyses so that the user is aware of the analysis performed to generate the estimates.



```
                              EPI-Meta
          Predicted Relative Risk with 95% Confidence Bounds
                    at User-Specified Dose Levels

Original Data File: C:\EPIMETA\EXAMPLES\EXAMPLE.REC

Within Study Model:
  Estimated Intercept, ln(Relative Risk), Dose/Exposure Level

Among Study Model:
  Random Effects:  Intercept =      -0.0665  Slope =       0.0098

  User-Specified           Predicted              95% Lower and Upper
   Dose Level            Relative Risk            Confidence Bounds
-----------------        -------------       -------------------------
      0.0000                0.9357          (      0.4881,        1.7937)
     25.0000                1.1968          (      0.3403,        4.2095)
     50.0000                1.5307          (      0.2175,       10.7720)
     75.0000                1.9579          (      0.1363,       28.1169)
    100.0000                2.5042          (      0.0848,       73.9287)

NOTE:  Dose/Exposure Levels are NOT Centered.
```

Figure 10. Results of User-Specified Predictions

Visually Guided Statistical Analysis: On the Representation, Use and Creation of Visual Statistical Strategies

Forrest W. Young
Psychometric Laboratory
University of North Carolina
Chapel Hill, NC, USA

David J. Lubinsky
Department of Computer Science
University of Witwatersrand
Johannesburg, South Africa

Abstract

The concept of statistical strategy is introduced and used to develop a structured graphical user interface for guided data analysis. The interface visually represents statistical strategies that are designed by expert data analysts to guide novices. The representation is an abstraction of the expert's concepts of the essence of a data analysis.

The interface consists of two interacting windows: the *guidemap* and the *workmap*. Each window contains a graph which has nodes and edges. The guidemap graph represents the statistical strategy for a specific statistical task (such as describing data). Nodes represent potential data-analysis actions that can be taken by the system. Edges represent potential actions that can be taken by the analyst. The guidemap graph exists prior to the data-analysis session, having been created by an expert. The workmap graph represents the complete history of all steps taken by the data analyst. It is constructed during the data-analysis session as a result of the analyst's actions. Workmap nodes represent datasets, data models, or data-analysis procedures which have been created or used by the analyst. Workmap edges represent the chronological sequence of the analyst's actions. One workmap node is high-lighted to indicate which statistical object is the focus of the strategy.

1.0 Motivation

Data are the lifeblood of science. Because computerized data-analysis systems help scientists understand data, they have become of central importance to the scientific enterprise, evolving into extensive and powerful systems capable of performing many kinds of very sophisticated and complex analyses.

Unfortunately, the structure of data-analysis systems has evolved willy-nilly over the years. While much thought has been focused on the kinds of analyses that can be performed by these systems, less thought has been given to their overall structure: It seems that the more powerful a statistical system is, the more clumsy it is to use.

In all statistical systems that we are familiar with, even when simple data-analysis procedures are used, novice users are soon at a loss as to how to combine several data-analysis procedures into a cogent statistical strategy that reveals the basic information in the data. The very power of many systems can actually hinder the data-analysis task, especially for users who are novices. We have the paradoxical situation that for many users, the increasingly powerful and sophisticated data-analysis systems are actually less suited to most users for understanding data.

Contact the first author for information or for software at forrest@unc.edu, or at the address above. We appreciate the many constructive discussions of the concepts and of their expression in this paper that we have had with Richard A. Faldowski.

In this paper we propose that data-analysis environments should support the visualization of statistical strategies and structures. We present an environment that guides the data-analysis steps taken by novice data analysts. Our environment also aids data analysts at all levels of sophistication by showing them the structure of their analysis session. In addition, sophisticated users can perform analyses simply by typing commands, if they don't want to use the graphical interface. Finally, our environment includes graphical tools that can be used by expert data analysts to create the analysis strategies that are used to guide novice analysts.

2.0 Background

We hold that data analysis is a highly complex activity (Young & Smith, 1991) that involves repetitive actions that occur over and over again. Thus, data analysis is a repetitive, cyclical search for understanding (Lubinsky & Pregibon, 1988). We believe that data analysis productivity, accuracy, accessibility and satisfaction will improve in an environment that guides and structures the actions that occur during the search for meaning in data.

One of our main design principles is that a data-analysis system should incorporate a variety of environments, each suited to a specific level of data analysis sophistication that a user might have, so as to maximize the data analyst's productivity and satisfaction. We believe that data-analysis software should be designed to accommodate the complete range of data analyst sophistication, from novice to expert.

We identify four kinds of data analysts: novice, competent, sophisticated and expert. Accordingly, we propose four kinds of environments: First, there should be *guidemaps* to guide novice data analysts through complete data analyses; second, there should be *workmaps* to inform novice and competent data analysts of the overall structure of their data-analysis sessions; third, there should be *command lines* to let sophisticated data analysts dispense with the visual aids when they find them unnecessary; finally, there should be an *authoring mode* to help expert data analysts create the guidance diagrams that are used by novices. In addition to these four environments, which are all highly interactive, there should be a script environment for automating repetitive data analyses. These five environments should be seamlessly integrated within the statistical analysis environment. Analysts should be able easily switch between them whenever desired, as we believe that analysts do not have the same level of expertise for all aspects of data analysis.

Structuring Data Analysis: Young & Smith (1991) argue that the process of data analysis is improved when the environment structures the actions taken by the data analyst. They suggest that an on-going data analysis should be represented by an icon-based graphical user interface which constructs a map of the analysis as it proceeds. This map shows the structure of the actions taken by the data analyst, and the data, models and analysis procedures involved in those actions. The map presents the analyst with a visualization of the structure of the analysis session, and can be used to return to previous steps.

For our work, the formal representation of session structure is the workmap. Our definition is: A ***workmap*** is a directed acyclic graph consisting of nodes and edges (as suggested by Young & Smith, 1991), where a node represents a data-analysis object (a dataset or a data model) or a data-analysis procedure that has been used by the analyst, and an edge represents the chronology sequence of the objects and procedure (the creation dependencies) during the analysis session. Taken as a whole, the workmap is a visual, object-oriented, directly manipulable, structured representation of the history of a data-analysis session.

Notice that a node is a self-contained unit of existing data (dataset), statistical computation (analysis procedure), or a combination of the two (data model), whereas edges represent the choices, actions and decisions that a data analyst made during the session. Nodes, which are the basic building blocks of the on-going data-analysis session, can be selected and reviewed at any time. The workmap visualizes the history of the on-going data analysis. It is a realization of a specific statistical strategy.

Guiding Data Analysis: At each step of a data analysis the data analyst is faced with many choices. Often, the data analyst returns to previous steps in order to make different choices. As stated by Lubinsky and Pregibon (1988), "Like a detective, a data analyst will experience many dead ends, retrace his steps, and explore many alternatives before settling on a single description of the evidence in front of him." We argue that data analysis will improve when it occurs in an environment that guides the actions taken by the analyst to understand data.

We use the Artificial Intelligence (AI) notion of strategy as a basis for developing methods for guiding data analysts. Several statisticians have developed the notion of a statistical strategy. These developments are extensively reviewed by Gale, Hand & Kelly (1993). Our definition of statistical strategy is: A ***statistical strategy*** is a formal representation of an expert statistician's conceptual structuring of 1) the data-analysis *procedures* to accomplish a specified data-analysis task; 2) the data analyst's *actions* (choices, decisions, etc.) that are possible with the procedures; and 3) the relationships between the procedures and actions needed to accomplish the task. The data-analysis task is to understand a specified data-analysis object (a dataset or data model).

For our data-analysis environment guidemaps are the formal representation of statistical strategy. Our definition is: A ***guidemap*** is a directed cyclic graph consisting of nodes and edges. The nodes of the graph represent data-analysis procedures, whereas the edges represent the analyst's possible actions. The structure of the map indicates the order dependencies between the procedures and the actions that can be taken with the procedures to accomplish the data-analysis task of understanding the data-analysis object. Finally, the data-analysis object (dataset or data model) is represented by a highlighted node of the *workmap*. It is said to be the focus object.

Notice that a guidemap node is a self-contained unit of *potential* statistical computation, while a guidemap edge represents the expert's guidance about moving from one computation to the next. Nodes are the basic building blocks of potential data analyses, i.e., of statistical strategies. On the other hand, the edges in the strategy represent the data analysts's possible choices, actions and decisions regarding the use of data-analysis procedures. They indicate permissible paths for traversing the nodes. Nodes can only be selected when they are highlighted. As a whole, the guidemap visualizes and abstracts the essence of an expert's statistical strategy.

3.0 Representing Statistical Strategy

In this section we discuss our definition of statistical strategy in detail, focusing on the four key aspects of the definition: the formal representation; the data-analysis object that is the focus of the strategy; the role of the expert statistician; and the objects, procedures and actions.

3.1 The Formal Representation of Strategy

First, our definition states that a statistical strategy is based on a *formal representation.* Our formal representation consists of graph structures like that shown in the guidemap window of Figure 1.This figure is a screen image from **ViSta**, the visual statistics research and development testbed (Young, 1994) that implements the ideas in this paper.

The guidemap, titled **Analysis Cycle**, presents the overall statistical strategy. This specific guidemap is always the first guidemap for a newly created dataset object. It is only a small portion of the overall strategy, since it causes addi-

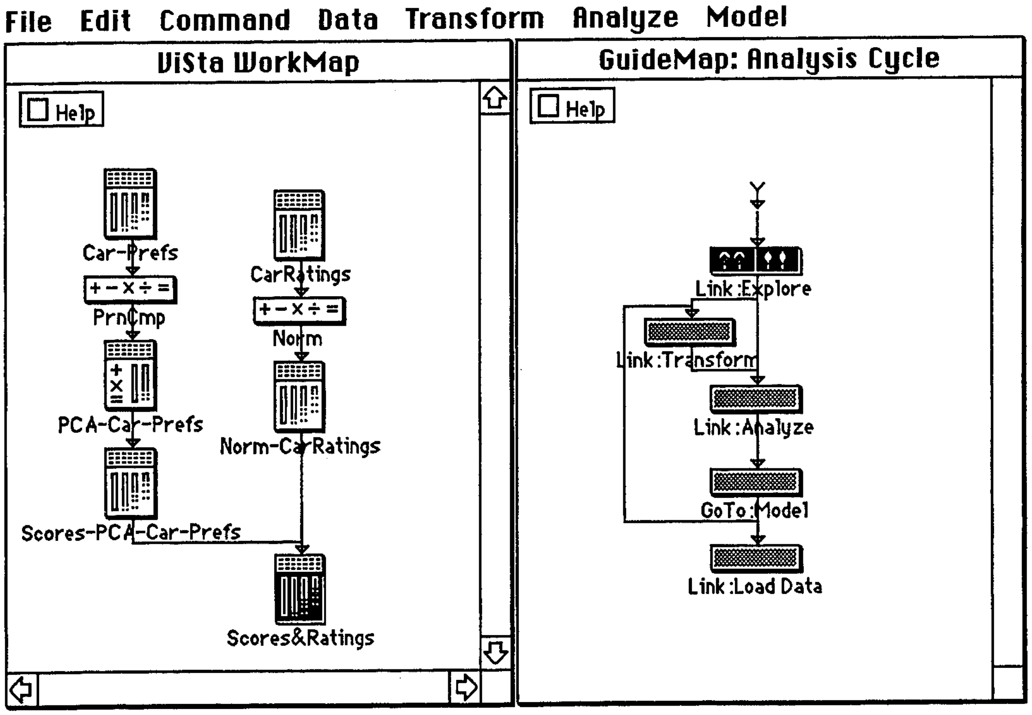


Figure 1: Formal Representation of Statistical Strategy in the WorkMap and GuideMap

tional "sub"-guidemaps to be displayed in the window. Taken as a whole, the guidemap in Figure 1, plus all of the additional guidemaps, are our formal representation of statistical strategy.

The strategy concerns a specific data or model object, thus, a data or model object is the focus of the analysis. The focus object is represented in the workmap window by the highlighted (dark) icon. The workmap itself shows where this object fits into the structure of the overall on-going analysis. The two separate windows emphasize the separation between the on-going data analysis (mapped in the workmap) and the strategy that is guiding the data analysis (mapped in the guidemap) We discuss the workmap in the next subsection. Here, we discuss the guidemap.

As stated above, the guidemap is a directed (possibly) cyclic graph consisting of edges and nodes. In our work, guidemap nodes are represented by the rectangular *button* icons, and guidemap edges are represented by the *arrows*. Thus, the buttons show potential steps in the analysis that the analyst is guided to take, whereas the arrows indicate the flow of guidance from one step to the next. A node is a self-contained unit of *potential* statistical computation which may do its own computations, or, recursively, call another strategy.

Buttons can be "active" or "inactive". Active buttons are highlighted (such as the Link:Explore button in Figure 1) and are ready to cause an action. Clicking on the ?? side of an active button enters a hypertext which causes help to be displayed about the action of the button. Clicking on the !! side of an active button enters a hypercode which causes the button's action to be initiated. Once the button's action has taken place, the high-lighting (activation) of the buttons changes: The clicked button deactivates, and the buttons that it points to are activated. Inactive buttons (such as the Link:Transform button in Figure 1) are not ready to do anything: Clicking on them has no effect.

There are two kinds of buttons: Flow Buttons, which control the flow between various portions of the large structure of guidemaps, and Procedure Buttons, which control the use of data-analysis procedures.

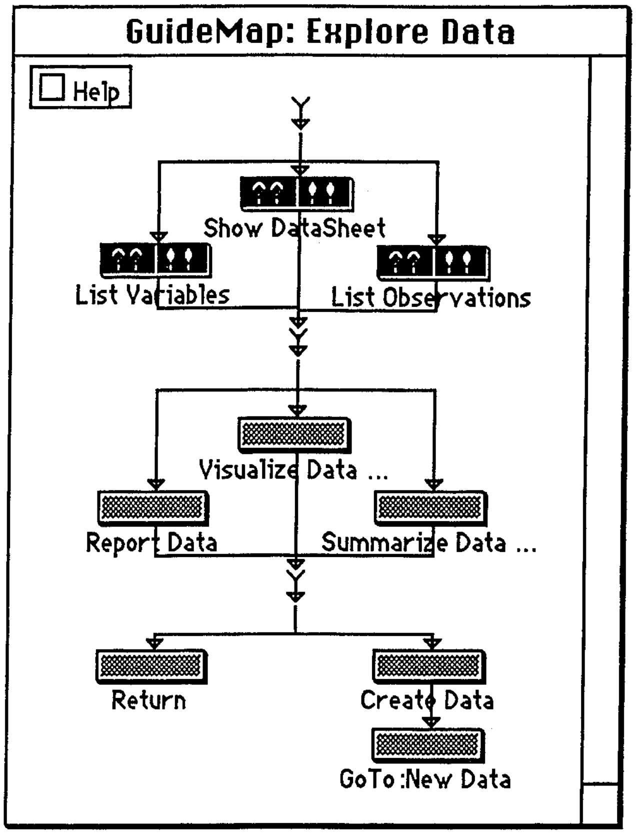


Figure 2: Formal Representation of Strategy for Exploring Data

Flow buttons include the Link, and GoTo buttons in Figure 1, and the Return button in Figure 2. These buttons take the user to other guidemaps. The Link button takes the analyst to a new strategy, whereas the Return button returns to the linked-from strategy. The Link button is, in essence, a *macro* data-analysis procedure which is itself a strategy, since this button opens up new strategies. For example, clicking on the !! portion of the Link:Explore button in Figure 1 causes the **Explore Data** guidemap, shown in Figure 2, to appear. Correspondingly, clicking on the Return button in Figure 2 (when it is highlighted) will take you back to Figure 1. Upon return to the guidemap in Figure 1, the high-lighting of the buttons will change according to the connecting arrows. That is, the Link:Explore button will de-activate, and the Link:Transform and Link:Analyze buttons will activate.

The GoTo button changes the focus of the data analysis, and of the strategy, to a new data or model object. When a new object has been created and named, then the name of that object replaces Data or Model in the GoTo button. Then, when the GoTo button is clicked, the appropriate data or model icon is highlighted in the workmap, and the appropriate strategy is displayed in the guidemap window.

All buttons other than flow buttons are procedure buttons that activate data-analysis procedures. In Figure 2 we see procedure buttons such as List Variables and Visualize Data. When an active procedure button is clicked, the indicated data-analysis procedure (listing variables, showing the datasheet) is activated.

3.2 The Focus of the Strategy

The focus of a statistical strategy is a data-analysis object (a dataset or a model). In Figure 1, the icons named CarRatings and Norm-CarRatings are data icons, whereas PCA-CarPrefs is a model. The focus object is represented by the icon that is highlighted in the workmap.

Each time a new object is created, it is represented by a new icon. Whenever a new dataset or model object is derived from an existing dataset object, an arrow is drawn from each of the new object's parents (usually only one) to the new object to show the creation dependency. These arrows have a meaning that parallels, but is somewhat different from, their meaning in the guidemap: They represent the flow of data into or out of a data-analysis object (dataset or model) or procedure as a result of a data analyst's action. In the guidemap, on the other hand, a arrows represent potential actions a data analyst might take.

The evolving progress of the data-analysis session is shown in the workmap. Certain actions taken via the guidemap create new nodes in the workmap. A new dataset object may be created by a mathematical procedure (such as normalization or principal components analysis) or by a non-mathematical operation (such as removing variables or merging datasets). A new model object is always created by a mathematical procedure. A procedure icon appears between the original and new objects when the creation involved mathematical operations, otherwise, no procedure icon appears. If a procedure icon appears, the creation dependency arrow is drawn from the parent objects through the procedure to the new object. Naturally, a new object may be brought in from "outside" of the system, in which case the new object is not connected to a parent (e.g., CarRatings in Figure 1).

The specific object which is the focus of the analysis (and, therefore, of the analytic strategy) is highlighted in the workmap. In Figure 1, Scores & Ratings is the focus object. Any data or model object in the workmap can be selected at any time to be the new focus object. When a new focus object is selected, the new strategy associated with it is displayed in the guidemap window, and the user enters that strategy.

The workmap and guidemap graphs differ in several respects. First, the structure of the guidemap graph doesn't change, it remains as shown throughout the analysis, although its high-lighting changes. The workmap graph, on the other hand, grows as new data and model objects are created and as new analysis procedures are used (both structure and high-lighting change). Second, the guidemap is a (potentially) cyclic graph, whereas the workmap is an acyclic hierarchical tree graph. This represents Lubinsky and Pregibon's (1988) observation that actions taken during data analysis are not hierarchical, but are cyclical, although the resulting analysis is hierarchical. Third, the guidemap (as represented by the initial guidemap shown in Figure 1, and all its sub-guidemaps) has an entry point but no exit point, whereas workmaps have both entries and exits. This represents the fact that a strategy has a beginning step but no final step. The lack of an exit point from a strategy reflects the fact that a strategy is cyclic, and that users should be able to quit a strategy (with the window's close box) whenever they choose.

3.3 The Role of the Expert Statistician

We turn now from the first two aspects of our definition of strategy (the *formal representation* and the *focus)* to the third aspect, namely that a statistical strategy represents the conceptual structure of an *expert statistician.*

It is assumed that the expert is only expert in a proscribed domain of statistical analysis, not for the entire domain. The role of such an expert is to decide, for the expert's area of statistical analysis expertise, what steps are involved, and in what order the steps should be taken. Thus, the representation shown in the guidemaps in Figures 1 and 2 (and in other guidemaps that are not shown) is on an experts knowledge about exploratory data analysis. These guidemaps represent the expert's conceptual structure of the sequence of steps involved in exploratory data analysis. The expert creates these guidemaps by using the "guidetools" that are discussed in Section 5.0.

3.4 The Objects, Procedures and Actions

The final aspect of our definition of strategy is that the expert's conceptual data analysis structure concerns three classes of things and the relationships among these things. The things are the *data-analysis objects*, the *data-analysis procedures*, and the *data analyst's actions*. All three are included in our representation of statistical strategy.

Data-analysis Objects: There are two types of data-analysis objects: dataset objects and model objects. Both types of data-analysis objects are represented by icons in the workmap (but not in the guidemap). Datasets are represented by tall rectangular icons containing very narrow vertical bars (representing variables). Models, like data, are represented by tall rectangular icons, but they contain mathematical symbols as well as "variable" bars to reflect the fact that models are data that have been subjected to mathematical operations. The highlighted data-analysis object is the focus of the statistical strategy.

Data-analysis Procedures: Procedures are represented by the wide rectangular icons in the workmap and guidemap windows. The procedures are the nodes of the guidemap's strategy structure, with each node being a self-contained piece of statistical computation, including visualizations (construction and presentation of dynamic statistical graphics), tables and textual results. These procedure-nodes, in the exploratory data-analysis example shown in Figure 2, include the show datasheet, list variables and list observations nodes, the data visualization, reporting, and summary nodes, and the node to create new data. These are the kinds of exploration procedures the expert deems to be appropriate parts of the analysis strategy.

Data Analyst's Actions: The possible actions of the data analyst are represented in the guidemap by the arrows connecting the procedure icons. On the other hand, in the workmap the arrows indicate actions that the data analyst has already taken. In the guidemap window, the direction of the arrow indicates the order in which the expert thinks the novice should use the data-analysis procedures. Thus, the data exploration strategy in Figure 2 indicates that the expert thinks the first three steps should be looking at the data themselves or listing their variable names or observation labels. Note that these procedure-buttons are highlighted and others are not. Once all three of these actions are taken, the next three buttons become highlighted (and the first three become gray), indicating that the next three analysis procedures are now available. In this way, the novice is guided through the data exploration strategy. At least one of the procedure-buttons in the guidemap window is always active, indicating which of the procedures can be used next by the analyst. Initially, when a strategy is entered, certain procedure(s) are highlighted, indicating what the analyst should do, and that the system is waiting for an action.

4.0 Using Statistical Strategies

In the previous section we described how we represent our concept of statistical strategy, a representation involving two graphs, called the guidemap and workmap. In this section we describe how the data analyst uses these two graphs.

4.1 Using the Guidemap

The guidemap window presents a map of an expert's statistical strategy. This map is used to guide data analyses performed by novice analysts. At the very beginning of the analysis of a new dataset object (see Figure 1), the guidemap window contains the **Analysis Cycle** guidemap. This guidemap presents the overall flow of a data analysis, emphasizing the major steps and their cyclical relationship. The initial highlighting of this map guides the user to explore the data, since the Link:Explore button is the only active (highlighted) button.

The flow of guidance is indicated by the arrows connecting buttons: When an active button's action is completed, the button deactivates (changes to gray), and the buttons that are pointed to by its arrows are activated. The change in high-lighting indicates the actions that the user is guided to take next, and the arrows indicate how guidance flows. Therefore, in Figure 1, after the data are explored the analyst is guided to transformation or analysis.

Lets consider how the guidemap in Figure 1 works. First of all, note that all of the buttons in the guidemap are *macro* buttons: Whenever one of them is used a new strategy map will replace the one shown in the figure. When the new strategy map is completed, the user will once again be shown the map in the figure, although it's pattern of high-lighting will have changed as indicated by the arrows. Thus, after exploring the data, the transformation and analysis (i.e., model fitting) buttons become highlighted. If transformation is chosen first, then when this is completed the analyst will be guided to analyze the data. If, instead, analysis is chosen before transformation, then when the analysis is complete the GoTo:Model button will become highlighted. Note, however, that if the Transform button was not used before the analysis, it will remain highlighted, so that the user now has the choice of either transforming the data and then re-analyzing, or of proceeding to look at the model. Finally, after looking at the model, the user can either transform the data once again, or start over with a new set of data. Thus, this map represents the expert's view that data analysis is a cycle that begins with exploration and which may or may not involve transformation before the first data analysis (model fitting). Then, the model resulting from the analysis should be looked at. The model may or may not suggest re-transformation, with this cycle of transformation, analysis and model inspection continuing indefinitely.

Note that when a new dataset object is created (for example, by transformation) the user will always be given the choice to change the focus of the strategy to the new data, thus beginning the analysis cycle all over again with a brand new, unused **Analysis Cycle** guidemap, starting with data exploration. On the other hand, the analyst may also continue focusing on the old data, if desired, although usually when new data are created the user will shift focus to them. Thus, there is an implicit cycle in the data-analysis process that does not appear in the guidemap: Whenever new data are created the analysis cycle usually recommences.

Let us now turn to consider what happens when the Link:Explore button is used. Since this button is a *macro* button (i.e., a button which corresponds to another guidemap), when it is used the map in the window changes to the **Explore Data** guidemap shown in Figure 2. Now, as indicated by the button highlighting, the analyst has the choice of three actions: show the datasheet, list variable names or list observation labels. When the user chooses any one of these three actions, the action takes place and the chosen button turns gray, since it is no longer a recommended action. The other two buttons remain highlighted.

Notice that the just-used button is connected to a short vertical arrow rather than to another button. This short vertical arrow is called an *and* icon because it is an "and gate" that restricts the flow of guidance from one action to the next. Specifically, all of the buttons that are connected *to* an *and* icon must be used before guidance can flow through the icon to the buttons that follow it.

Thus, when one of the active buttons in Figure 2 is used, no other buttons become highlighted until all three active buttons are used. Then, all of the buttons that have arrows pointing to them from the *and* icon are activated. In this way the user is guided to use all three active buttons in Figure 2 before doing anything else. They can be used in any order. Once they are all used, the next group of three buttons is activated, and the analyst must use them (in any order) before going on. After these three buttons have been used, the map appears as shown in Figure 3.

The guidemap in Figure 3 has changed from the one in Figure 2: The data analyst is now being guided to either return to the guidemap which led to this one (the one shown in Figure 1, but with the Transform and Analyze buttons activated) or to create a new dataset object. The analyst may wish to take the latter step to create a subset of the original data. If the decision is made to create new data, then the analyst has the choice of going to those data, which brings up a brand new **Analysis Cycle** map (identical to that shown in Figure 1) or of returning to the old **Analysis Cycle** map (with the structure shown in Figure 1, but with Transform and Analyze activated).

Note how the strategy has guided the analyst: As shown in Figure 1, the analyst must explore the data first. The analyst must analyze the data before inspecting the model. In Figure 2 and 3 the analyst must look at the data and their identifying information before visualizing the data or getting summary statistics. On the other hand, the data analyst has choices: In Figure 1, it is not required, though it is possible, to transform the data before fitting the model. Similarly, in Figure 2, it is possible to visualize the data before seeing summary statistics, or to do the actions in the reverse order.

4.2 Using the WorkMap

In the example shown in Figure 1, the workmap shows a data-analysis session that has already involved several major steps. In the first step, the analyst read in the data that defined the CarPrefs dataset object. These data were then submitted to a Principal Components Analysis, as indicated by the PrnCmp procedure icon. This analysis produced the PCA-CarPrefs model object. The analyst then requested that a new dataset object Scores-PCA-

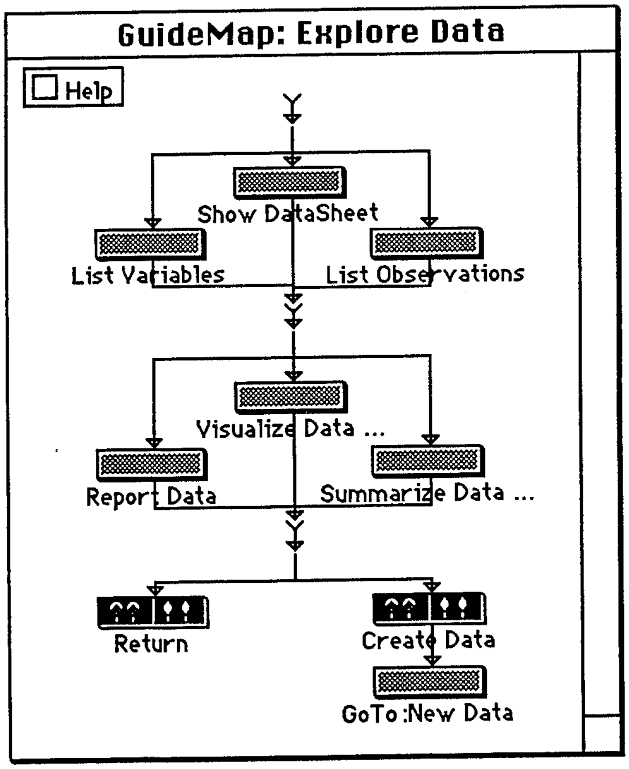


Figure 3: Strategy for Exploring Data after using several analysis procedures.

CarPrefs be created by the model object. Separately, the analyst also read in data that defined the CarRatings dataset object. These data were normalized, as indicated by the Norm procedure icon, creating a new dataset object named Norm-CarRatings, which was merged with the Scores-PCA-CarPrefs dataset object to obtain another dataset object named Scores & Ratings (the current focus of the statistical strategy).

It should be emphasized that portions (or all) of the data analysis can be created directly in the workmap window, without using the guidemap window, whenever a sufficiently sophisticated data analyst wishes. An entire data analyses can be created from the workmap without ever seeing a guidemap.This can be done by clicking the mouse on the body of an icon to obtain a pop-up menu of actions that the icon supports. These menu-items are also accessible from the **Data** and **Transform** menus of the menubar shown at the top of Figure 1. The pop-up menu for model icons corresponds to the **Model** menu in the menubar. The analysis procedures are accessed from the menubar's **Analyze** menu (and from an optional workmap toolbar that is now shown).

It should also be emphasized that a previous portion of the data analysis can be revisited at any time by simply clicking on the appropriate workmap icon. Then, the analysis can be continued in a new direction by simply taking different steps than were taken previously. Thus, the workmap graph provides a very convenient and simple way of backtracking, a feature that can be very hard with conventional systems which do not keep a full history of a data analysis session. This can be done across sessions by saving (portions of) the workmap and reloading it in a later session (only partially implemented at this time).

Also, note that if the data analyst is performing the analysis directly from the workmap guidance is available at anytime by simply requesting that the guidemap be shown. When so requested, the appropriate portion of the guidemap structure is displayed in the guidemap window. Thus, it is possible for the data analyst to use guidance when needed, and to avoid it when it is not needed.

5.0 Creating Statistical Strategies

The guidemaps that embody statistical strategy are created while in "authoring" mode. In this mode there is an **Author's WorkBench** window in which new guidemaps are created. In addition, a **Tools** menu is added to the menubar, and the action of all **Data, Transform, Analyze** and **Model** menu items is enhanced.

Taken together, the modified menu items and the new **Tools** menu items are "guidetools" that are used to create new guidemaps.The expert uses these guidetools to create the buttons that are to become the nodes of the guidemap. Recall that there are flow buttons, which control the flow between portions of the analysis, and procedure buttons, which control the use of data-analysis procedures. The **Tools** menu creates flow buttons, while the other menus create procedure buttons.

Procedure buttons are created by using those menu items that are needed to perform the specific type of data analysis for which guidance is being created. When in authoring mode, the action of the menu items is modified so that, in addition to the analysis action taking place, a button is placed on the author's workbench (the button's title is the same as the menu item's name).

Note the basic design philosophy underlying the creation of statistical strategies: The expert creates the guidemap's data-analysis procedure buttons by using the menu system in exactly the same way that s/he would use it when it is not in authoring mode. Since the system is in authoring mode, buttons appear in the workbench window. Otherwise, everything is the same as when the system is not in authoring mode. This design feature means that the expert is free to perform whatever analysis is desired, using whatever data-procedures are appropriate, without any new authoring "features" changing the process.

On the other hand, flow buttons, which do not correspond to data-analysis actions, are created by using the new authoring "features" that are represented by items of the **Tools** menu. There is a menu item for each type of flow button, including **Link, GoTo, Return** and **And** items (for icons shown in previous figures), **AutoLink** and **AutoReturn** items that cause a guidemap to automatically link to another guidemap and to automatically return to the linked-from guidemap, and an **Initial** item to indicate which buttons are to be activated when the guidemap is initially displayed. Thus, while the author does not need to learn any new aspects of the system while creating the procedure steps of the data analysis, new features must be learned to indicate flow control (the actual guidance). In a more complete implementation, many additional flow-control features would be available.

Once the expert has placed two or more buttons or icons on the workbench, s/he can connect them together with an arrow drawing tool. Of course, at any time the buttons and icons can be dragged to new locations to give the guidemap a more pleasing and comprehensible layout. The arrows automatically reposition themselves to reflect the new layout. Of course, when the map is entirely created, the expert saves it for later use by the novice.

Finally, the expert must create the help information that is displayed when the novice clicks on the ?? side of a button. This is done by using an ordinary text editor, and by saving files with certain naming conventions so that they can be found when needed.

6.0 Discussion

In this section we discuss the relation of our work to hypertext and to visual programming, two concepts with their origins in computer science.

6.1 Hypertext and Hypercode

Hypertext (or, more generally, hypermedia) is a generic approach to linking and structuring all forms of computerized materials so that non-linear, dynamic documents can be constructed (for more information, consult Woodhead (1990) or Martin (1990)). Hypermedia consist of nodes that are connected by links. The nodes contain the materials, which may be text, diagrams, animations, images, video, sound, computer programs or any other computerized information. The links provide a mechanism for non-linear navigation among the nodes. The nodes may be linked together into web, hierarchical, cyclic, or other structures. Hypermedia always have tools for navigating the link structure and for displaying the node material.

Clearly, our help system is a hypertext: Guidemap buttons are nodes that contain help text, and arrows are links between nodes. In addition, the ?? side of a guidemap button is the tool that accesses and displays the hypertext. The buttons also navigate the hypertext. Finally, the structure of the hypertext is shown by the structure of the guidemap.

Of much more interest is the fact that our guidance system is a "hypercode", a form of hypermedia where the materials are computer programs. Note that the structure of the hypercode is represented by the structure of the guidemap, and that the hypercode is navigated by clicking on the !! side of guidemap buttons. When the naive analyst clicks on the !! side of a button, the button not only navigates to a particular piece of hypercode, but also causes the execution of that piece of code. Thus, from the point-of-view of the naive user, the guidemaps display the structure of the guidance hypercode, provide a means of navigating through it, and a means of executing pieces of it. (Note that the guidemaps also display the structure of the *help* hypertext, provide a means of navigating through it, and for displaying pieces of it. Thus, both the hypertext and hypercode are seamlessly unified.)

It follows that the expert user's process of authoring guidemaps is, in fact, a process for writing hypercode. As described above, authoring involves creating two kinds of buttons: action buttons and flow buttons. When an action button is created, the code that is written is a ViSta function which parallels a data-analysis menu item and which causes a data-analysis step to take place. On the other hand, when the author creates a flow button, the code that is written consists of standard Lisp flow control functions.

Thus, authoring guidemaps is computer programming. However, it is not the usual type of programming in which the programmer types statements. Rather, it is one in which the statements get generated automatically when the author (programmer) selects a button. This form of computer programming is known as visual programming, which is discussed in the next section.

6.2 Visual Programming & Program Visualization

Visual programming and program visualization are very active areas of research in computer science. There goal is to simplify programming, and to make programming accessible to a wider audience. They attempt to reach this goal by combining the disciplines of interactive graphics, computer languages and software engineering to take advantage of a person's non-verbal visual capabilities and a computer's interactive graphical capabilities.

Conventional textual computer languages process program instructions that exists in one-dimensional, nongraphical (textual) streams. Visual programming, by contrast, refers to a way for people to create programs using graphical methods. These icons can be viewed as two-dimensional graphical instructions (Myers, 1990), as opposed to one-dimensional textual instructions (although the two-dimensional visual program is translated into an underlying one-dimensional textual program).

Program visualization, on the other hand, is an entirely different concept: Here, the program is specified in the usual textual manner, but is then illustrated visually in some form. Thus, the program is specified as text and translated into graphics. Note that this reverses the process involved in visual programming, where the program is specified as graphics and is translated into text.

Guidemaps and workmaps are simple examples of visual programming and program visualization. Guidemaps are visual programs which have been created by an expert using a visual authoring system, and which are "executed" by the novice. Workmaps are program visualizations which have been created textually (or visually). In fact, when a workmap is saved and re-executed, it becomes a visual program as well as a program visualization.

The earliest visual languages were computerized flowcharts. More recently, visual languages are formally based on graph theory, consisting of nodes and edges (note the connection with hypertext). Often the edges are directed (and called arrows). There are graphs such as "higraphs", which allow nodes to contain other nodes and which permit arrows to split and join, or "colored petri nets" which allow parallel processing systems to be constructed. A number of visual programming systems use dataflow diagrams. Here the operations are typically put in nodes, and the data flow along the arrows connecting the nodes.

We have based guidemaps on directed cyclic graphs and workmaps on directed acyclic dataflow diagrams (Young & Smith, 1991). Our developments are limited, however, in that we have not developed looping or conditional branching. Thus, one can argue that our workmaps and guidemaps do not constitute a full visual programming language, since the abstract definition of a computer language requires the inclusion of these capabilities.

We recommend investigating the feasibility of developing (or using an existing) visual dataflow language as the basis for a structured graphical interface for performing and guiding data analysis. Two interesting existing systems are VisaVis (Poswig, Vrankar & Morara, 1994) and Khoros (Rasure & Williams, 1991). Both are functional visual programming languages with looping and conditional branching. Khoros is also a dataflow language.

7.0 Conclusion

Understanding and representing statistical strategy is a relatively new area of research that is just now gaining momentum. Within this area of research, it appears that our visual approach to statistical strategy is new and unique, and is firmly based on current computer science thinking. As the capability of computers continues to increase, while their price continues to decrease, the audience for complex software systems such as data-analysis systems will become wider and more naive. Thus, it is imperative that these systems be designed to guide data analysts who need the guidance, while at the same time be able to provide full data-analysis power. An efficacious way of doing this is certainly needed, and we believe that our visualized statistical strategies have the potential for great payoff in the improvement of the quality, satisfaction and productivity of statistical data analysis.

Naturally, we hope that our visual methods for guiding naive data analysts by visually representing, using and creating statistical strategies will prove useful. Of much greater importance, however, is our basic point: Concentrated attention should be given by computational statisticians to the representation, usage and creation of statistical strategies. We believe that such strategies should be available to guide and structure the data-analysis process so that relatively naive users can perform high-quality data analyses. And we believe that guidance systems should be empirically tested to see if they deliver on their promise.

8.0 References

1. Gale, W.A., Hand, D.J. & Kelly, A.E. (1993) Statistical Applications of Artificial Intelligence. In: C.R. Rao, *Handbook of Statistics: Computational Statistics,* Amsterdam: Elsevier North-Holland, **9**, 535-576.
2. Lubinsky, D.J. & Pregibon, D. (1988) Data Analysis as Search. *Journal of Econometrics*, **38**, 247-268.
3. Martin, J. (1990) *Hyperdocuments and how to create them.* Prentice Hall, Englewood Cliffs, NJ
4. Myers, B.A. (1990) Taxonomies of Visual Programming and Program Visualization. *Journal of Visual Languages and Computing*, **1**, 97-123.
5. Poswig, J., Vrankar, G. & Morara, C. (1994) VisaVis: A Higher-order Functional Visual Programming Language. *J. Visual Languages and Computing*, **5**, 83-111.
6. Rasure, J.R. & Williams, C.S. (1991) An Integrated Data Flow Visual Language and Software Development Environment. *Journal of Visual Languages and Computing*, **2**, 217-246.
7. Woodhead, N. (1990) *Hypertext & Hypermedia: Theory and Applications.* Addison-Wesley. New York.
8. Young, F.W. (1994) ViSta - The Visual Statistics System. *Psychometric Lab Report 94-1.* UNC Psychometrics Lab, Chapel Hill, NC.
9. Young, F.W. and Smith, J.B. (1991) Towards a Structured Data Analysis Environment: A Cognition-Based Design. In: Buja, A. & Tukey, P.A. (Eds.) *Computing and Graphics in Statistics*, 36, 253-279. New York: Springer-Verlag.

Visually Guided Multidimensional Scaling with ViSta-MDS

Mary McFarlane
Bowman Gray School of Medicine,
Wake Forest University

Abstract

This paper demonstrates the ability of ViSta-MDS (Young, 1994; McFarlane, 1992) to facilitate guided data analysis in the framework of multidimensional scaling. The paper begins with a brief description of the goals of a multidimensional scaling analysis. Next, the guidance properties of the ViSta-MDS module are described. The paper illustrates each step in a guided data analysis, and shows many of the GuideMaps encountered along the way.

1.0 Multidimensional Scaling

The data for a multidimensional scaling analysis are in the form of dissimilarity matrices, e.g., a group of judges rates the dissimilarity between a number of stimuli. Each stimulus in the set is compared with every other stimulus to form a matrix of dissimilarity judgments. Each judge contributes one matrix of dissimilarities to the data set.

The goal of a multidimensional scaling (MDS) analysis is to produce a low-dimensional solution space such that the Euclidean distances between the points in the solution space most closely approximate the dissimilarity judgments provided by the judges. A popular measure of fit in multidimensional scaling analyses is stress, defined as the square root of the sum of the squared differences between the dissimilarity judgments provided by the judges and the Euclidean distances in the MDS solution space.

2.0 Guided Statistical Analysis

As is true with many statistical models, users of multidimensional scaling are often not familiar with the techniques and assumptions associated with such an analysis. In order to accommodate the wide variety of users, we have developed a guided statistical analysis system in which expert users may provide guidance for less experienced or novice analysts. In this system, expert users create a statistical strategy for novice users to follow. The strategy may be represented either graphically or in a text file; for the purposes of this exposition, we will focus on the graphical representation of the expert's statistical strategy, called the GuideMap.

A user begins a guided data analysis session by selecting "Show GuideMap" from either the "Command" menu or the WorkMap; the first GuideMap, shown in Figure 1, appears. As described in more detail by Young & Lubinsky (1994), a GuideMap consists of buttons which can be used to carry out steps in the analysis. The initial guidemap is very simple: It simply prompts the user to load data. If the user clicks on the left half of the "Load Data" icon (on the ??), a help screen appears and the user is given information about the loading of data in ViSta. If the user clicks in the right half of the icon (on the !!), loading a data file is initiated. All guidemap buttons can provide help about an analysis step and can cause the step to be taken.

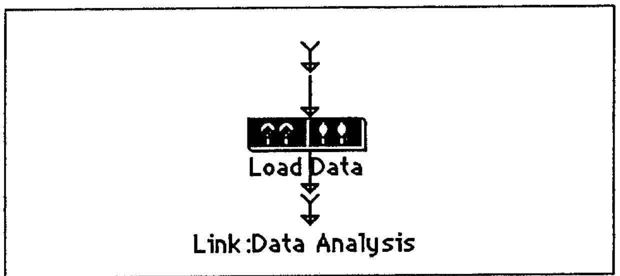


FIGURE 1. The initial ViSta GuideMap prompts the user to load data.

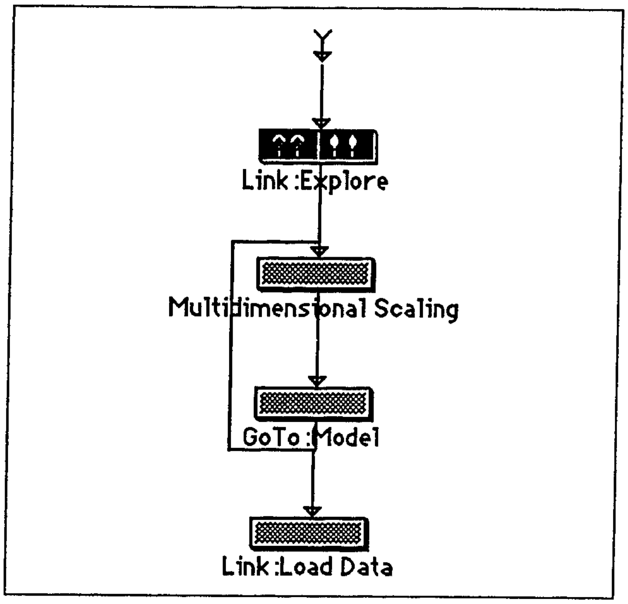


FIGURE 2. The Dissimilarity Analysis GuideMap guides the user to explore the data.

When the user selects a dissimilarity data set to be loaded, the GuideMap is updated, becoming the one shown in Figure 2. As is the case for most guidemaps, this one has several buttons. Some of these buttons are "active" (the dark, highlighted ones), whereas some are "inactive" (the gray ones). Active buttons (such as the "Link:Explore" button in Figure 2) are ready to cause an action. Once the button's action has taken place, the highlighting (activation) of the buttons changes: The clicked button deactivates, and the button(s) to which it points are activated. Inactive buttons (such as the "Multidimensional Scaling" button in Figure 2) are not available for any action: Clicking on them has no effect.

The GuideMap in Figure 2 prompts the user to explore the data. Clicking the "Link: Explore" button causes the GuideMap that guides users through a data exploration to appear. This GuideMap is shown in Figure 3. This GuideMap has three sections: First, the user is guided to examine the datasheet, list the variables, and list the observations. Note that only these three buttons are highlighted, so only these actions can take place. Once all three of these actions occur, the next three buttons are highlighted, indicating that the user is now guided to visualize the data, to get a data report, or to compute summary statistics. Finally, when the user has used these three buttons, the "Return" and "Create Data" buttons are available, permitting the user to return to the previous guidemap (the one shown in Figure 2, but with the highlighting changed) or to create new data.

Let us examine the data-visualization step in greater detail. Though the visualization of the multidimensional scaling solution space is more informative than the visualization of the data, the data-visualization step often results in interesting revelations. Figure 4 shows the ViSta-MDS data-visualization screen. The ratings from each judge are plotted against the ratings of every other judge in the Scatterplot Matrix at the upper left of the screen. The Scatterplot Matrix serves as a control panel for the visualization in the other plots; clicking on any cell of the Scatterplot Matrix causes a larger version of that cell to appear in the Scatterplot at the lower left. Clicking on any two cells in the same row or column of the Scatterplot Matrix causes the three dimensions common to the two cells to appear in the SpinPlot at the upper center of the screen. Finally, the Histogram at the bottom center of the screen reflects the ratings provided by the judge represented by the row of the currently selected cell of the Scatterplot Matrix. By examining the Histogram, the user can determine whether a particular judge is inclined to give extreme, possibly biased, dissimilarity ratings. By examining the higher-dimensional plots, the user may better understand the degree to which judges agree with each other.

After the data are visualized, the GuideMap shown in Figure 3 prompts the user to Report Data and Summarize Data. A click on the Report Data icon produces a text

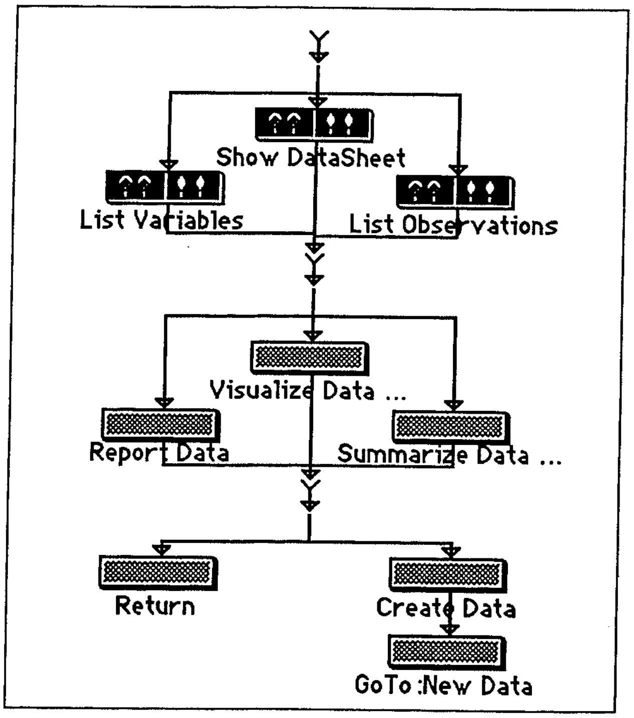


FIGURE 3. The data-exploration GuideMap prompts the user to explore the data both graphically and textually.

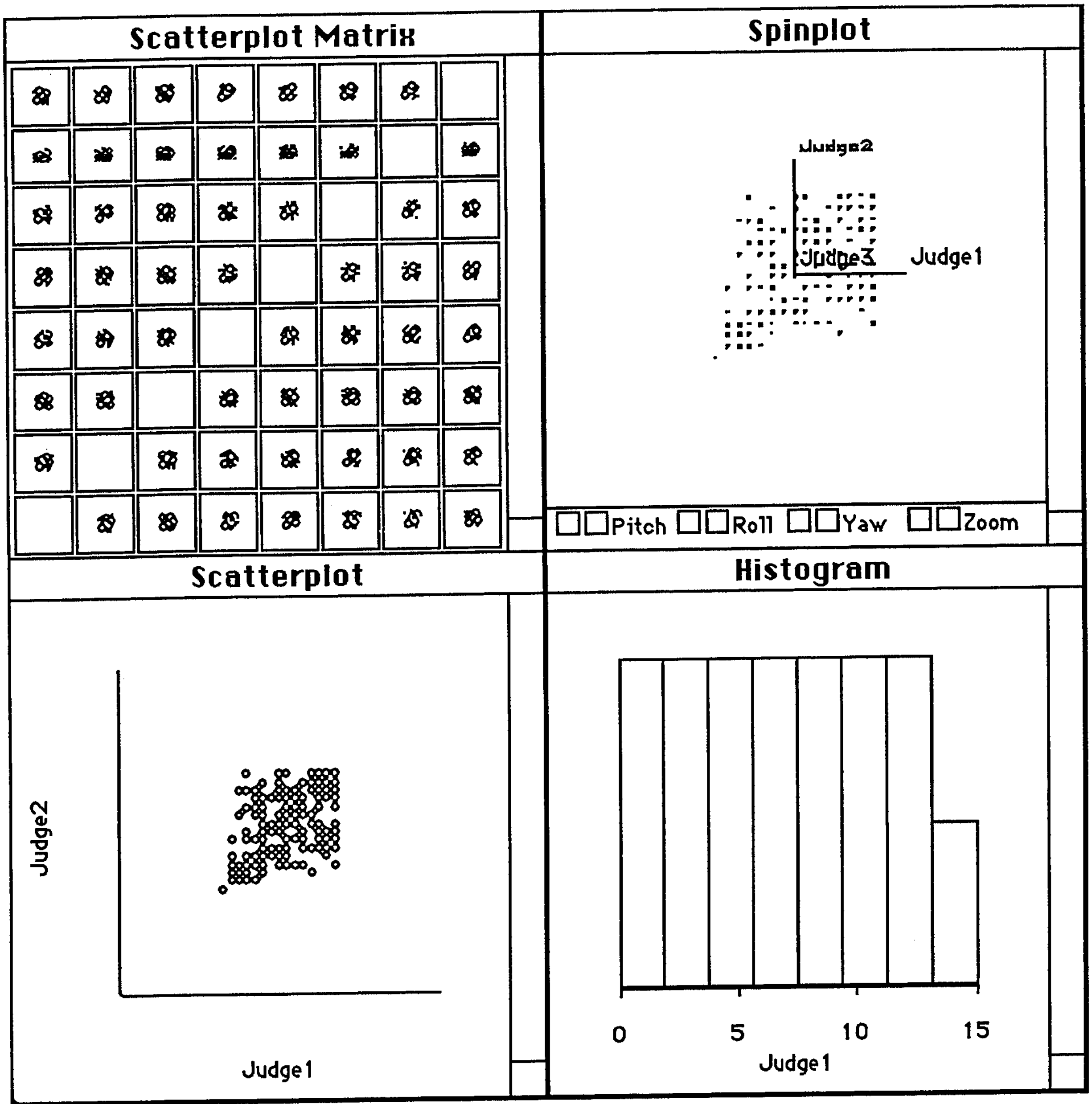


FIGURE 4. The spreadplot provides information about each judge in the data set. Judges' ratings may be viewed individually or may be compared with other judges' ratings.

screen showing each matrix in the data set, labelled by judge, and the stimuli on which the ratings were given. A click on the Summarize Data icon produces standard summary statistics such as mean, variance, skewness, kurtosis, range, and quartiles for each matrix in the data set. These summary statistics provide the user with some knowledge as to the rating style of each judge. By presenting both graphical and textual displays of this information, ViSta-MDS facilitates understanding of multidimensional scaling data by a wide range of users.

After visualizing, summarizing and reporting the data, the GuideMap for exploring data looks like the one shown in Figure 5. The button highlighted now guides the user to either return to the GuideMap which led to this one or to create data from a subset of the current data. By choosing the "Return" option, the user returns to the GuideMap shown in Figure 6. This GuideMap guides the user to perform a multidimensional scaling analysis. It is important to realize that the expert author of the GuideMap has already selected desirable options for a multidimensional scaling analysis; thus, when a novice clicks on the Multidimensional Scaling button, the computations are carried

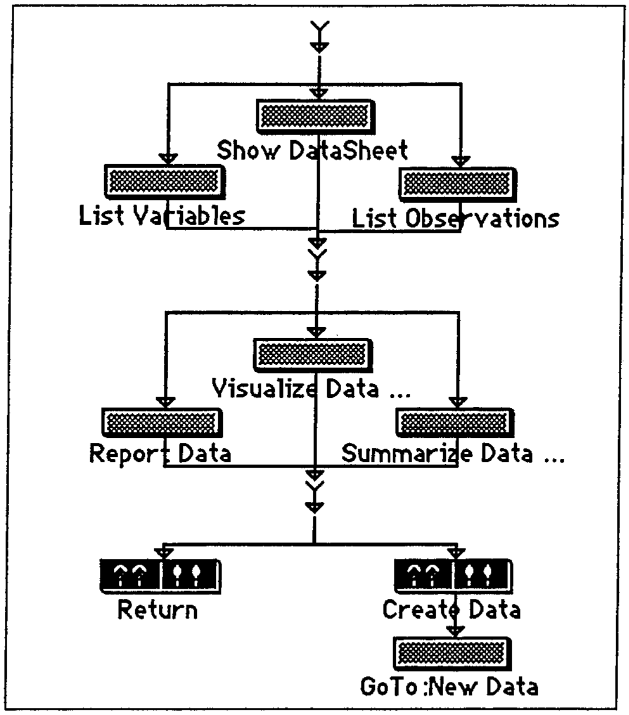


FIGURE 5. The GuideMap for exploring data after the data have been explored.

out with the set of options provided by that expert. The novice effectively places the decisions about the options in the analysis in the hands of the expert author of the GuideMap. When the analysis is complete, the highlighting of the GuideMap in Figure 6 changes so that the user is prompted to "Goto: Model". When this button is clicked, the GuideMap for modeling data (Figure 7) appears.

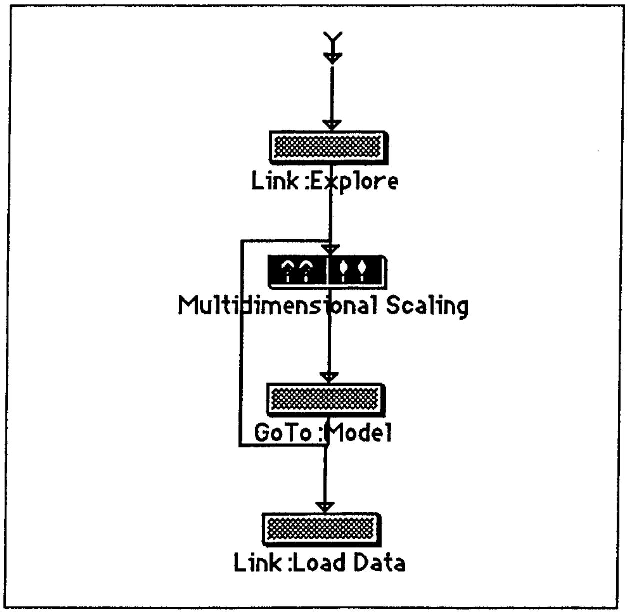


FIGURE 6. The basic data analysis guidemap, after the data are explored, now guides the user to perform the Multidimensional Scaling.

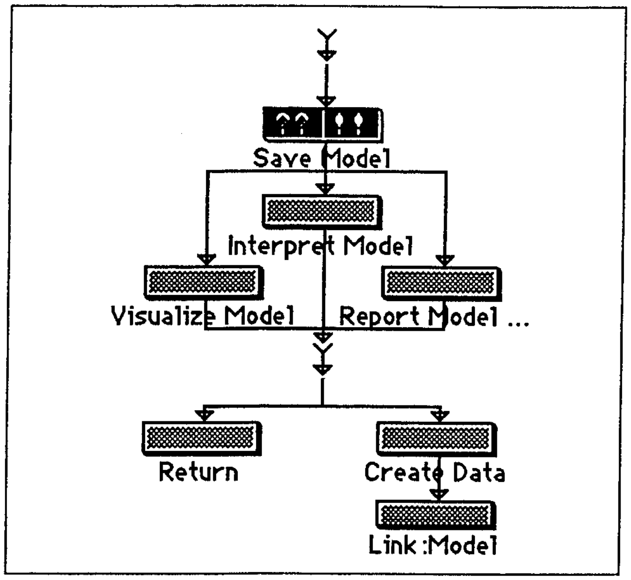


FIGURE 7. The model-exploration GuideMap prompts the user to explore the multidimensional scaling model both graphically and textually.

In this GuideMap, the user is first required to save the current model; next, the Interpret Model button must be clicked. This produces a text window that contains a description of the various components of the multidimensional scaling model, and a brief summary of the best way to examine and interpret those components. In order to examine the components of the multidimensional scaling model described in the Interpret Model screen, the user must use the Visualize Model and Report Model buttons.

The Visualize Model button produces the spreadplot shown in Figure 8. The Scatterplot Matrix shows each of the dimensions of the solution space plotted against every other dimension of the space. The Scatterplot Matrix is the control panel for the Stimulus Plane and Stimulus Space plots in a manner analogous to that of the Scatterplot Matrix, Scatterplot and SpinPlot in the data-visualization screen. The visualization of the model includes a scree plot, showing the variance accounted for by each dimension in the multidimensional scaling solution space, with a vertical line indicating the dimensionality of the current model. The Stress Plot at the lower right of the screen shows the value of the stress index for the current model. This index may be optimized by clicking the "Iterate" button in the Stimulus Space plot, as described in McFarlane and Young (1994).

The Report Model icon produces a text screen that provides information about the multidimensional scaling model. The analyzed matrix is shown, along with the additive constant required to make that matrix positive-definite. The initial stimulus coordinates are shown, followed by the current stimulus coordinates, which reflect changes made by either iteration or visual sensitivity analysis (McFarlane and Young, 1994). The current value of the stress index is also reported. Again, ViSta-MDS provides both graphical and textual displays of information to enhance the understanding of users at all levels of expertise.

3.0 Conclusion

ViSta-MDS is a testbed for visual statistical analysis that is still under development. The goal of the software is to provide novice, sophisticated and expert users the necessary guidance to perform appropriate statistical analyses. This goal is reached through the use of GuideMaps and WorkMaps that provide both graphical and textual displays of information. ViSta-MDS also facilitates visual sensitivity analysis of the multidimensional solution space, as described in McFarlane and Young (1994). It is hoped that the guidance and interactive graphical capabilities provided by ViSta-MDS will lead to the enhanced understanding of multidimensional scaling analysis by a variety of users.

4.0 References

1. McFarlane, M.M. (1992) Interactive graphical modeling for multidimensional scaling. University of North Carolina at Chapel Hill Department of Psychology. Unpublished master's thesis.
2. McFarlane, M.M. & Young, F.W. (1994) Graphical sensitivity analysis for multidimensional scaling. J. of Computational and Graphical Statistics, 3, 23-34.
3. Young, F.W. (1994) ViSta - The Visual Statistics System. *Psychometric Lab Report 94-1.* UNC Psychometrics Lab, Chapel Hill, NC.
4. Young, F.W. & Lubinsky, D. (1994) Visually Guided Statistical Analysis: On the Representation, Use and Creation of Visual Statistical Strategies.

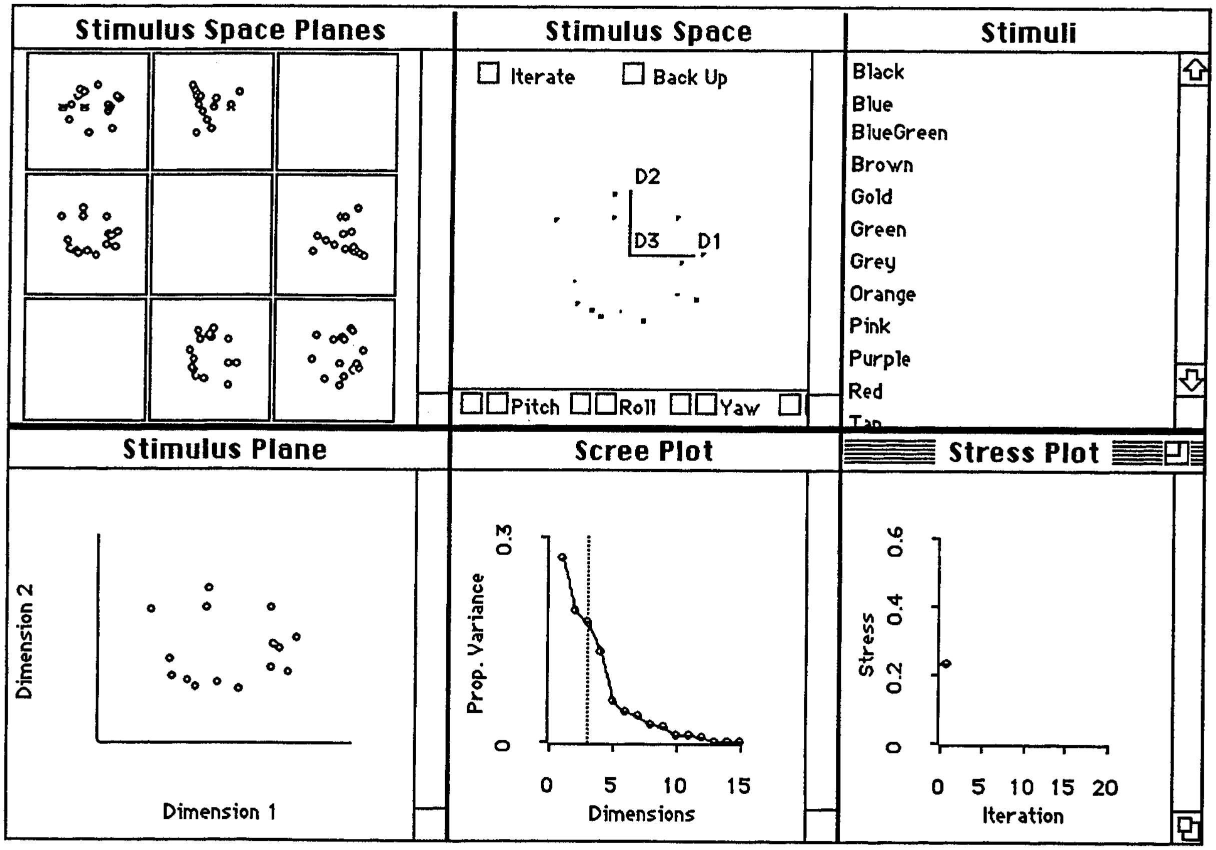


FIGURE 8. Model visualization in ViSta-MDS.

Visual Correspondence Analysis

Bee-Leng Lee
Department of Economics and Statistics
National University of Singapore
Singapore

Forrest W. Young
Psychometric Laboratory
University of North Carolina
Chapel Hill, NC, USA

Abstract: In this paper we describe a new statistical environment for correspondence analysis which incorporates the traditional analysis methods with dynamic graphical procedures. We make use of algebraically linked plots to visualize the solution space and the quality of representation under various dimensions. We also introduce interactive graphical modeling as a complementary tool to the traditional algebraic analysis, which allows the data analyst to modify the configuration of points and to examine the resultant effect.

1 Introduction

Correspondence analysis has been used primarily to analyze two-way contingency tables, in which the observed associations of two categorical variables are summarized by the cell frequencies. The name is a translation of the French *Analyses des Correspondances*, where the term *correspondances* denotes a "system of associations" between the elements of the data.

In essence, correspondence analysis performs a form of perceptual mapping similar to multidimensional scaling, where the categories are represented as a set of row and column points in the multidimensional space, and proximity indicates the level of association among the row or column categories. The objective is to represent the inter-point distances in a smaller dimensional subspace—such that the original distances are preserved as much as possible—for ease of visualization.

To illustrate correspondence analysis, consider the multidimensional time series on the number of science doctorates conferred in the USA from 1960 to 1975 that is shown in Table 1 (Greenacre, 1984). Correspondence analysis of these data yields the graphical display shown in Figure 1.

In Figure 1, there are two sets of points, as indicated by the two types of point symbols. The points are *row* points for the 12 disciplines (represented by crosses) and *column* points for the 8 years (represented by disks). Distances between points within the same set (row-to-row and column-to-column) are defined in terms of chi-square distances, which can be interpreted as a measure of similarity between the frequency profiles. For example, the anthropology degree and the engineering degree are far from each other because their profiles are different, whereas the mathematics degree is near the engineering degree because their profiles are similar. On the other hand, distances between points of different sets (row-to-column) do not approximate any defined quantity and are not directly comparable. The interpretation of such distances is governed by the barycentric relationship between the rows and columns (Greenacre and Hastie, 1987). In this example, each discipline point lies in the neighbourhood of the year in which the discipline's profile is prominent. Thus, there are relatively more chemistry and agriculture degrees in 1960, while the trend from 1965 to 1975 appears to be away from the physical sciences.

A new statistical environment for correspondence analysis has been created in ViSta (Young, 1994), called ViSta-CA, which incorporates the traditional analysis methods of

Contact the second author for information or for software at forrest@unc.edu, or at the address above.

correspondence analysis with graphical procedures. Results of correspondence analysis are presented visually via dynamic statistical graphics, the purpose being to help the analyst visually explore the structure of the geometric model.

2 Algorithm

Let $\mathbf{X}$ be an $(n \times m)$ matrix of observed frequencies of rank q such that the row sums and column sums are nonzero. Let $\mathbf{1}$ be a row vector of ones and $\mathbf{I}$ be an identity matrix, each of appropriate orders. Denote a matrix-valued function that creates a diagonal matrix from a vector by *diag*( ). Define

i. $s = \mathbf{1}'\mathbf{X}\mathbf{1}$ as the sum of all elements in $\mathbf{X}$;

ii. $\mathbf{P} = \frac{1}{s}\mathbf{X}$ as the matrix of relative frequencies;

iii. $\mathbf{r} = \mathbf{P}\mathbf{1}$ as the vector of row masses;

iv. $\mathbf{c} = \mathbf{P}'\mathbf{1}$ as the vector of column masses;

v. $\mathbf{D}_r = diag(\mathbf{r})$ as a diagonal matrix of row masses; and

vi. $\mathbf{D}_c = diag(\mathbf{c})$ as a diagonal matrix of column masses.

The generalized singular value decomposition (abbreviated SVD) of $\mathbf{P}$ provides the required solution to the point coordinates of correspondence analysis:

$$\mathbf{P} = \mathbf{A}\mathbf{D}_u\mathbf{B}'$$

where

i. $\mathbf{A}$ is an $(n \times q)$ matrix whose columns are the left generalized singular vectors;

ii. $\mathbf{D}_u$ is a $(q \times q)$ diagonal matrix of generalized singular values;

iii. $\mathbf{B}$ is an $(m \times q)$ matrix whose columns are the right generalized singular vectors; and

iv. $\mathbf{A}'\mathbf{D}_r^{-1}\mathbf{A} = \mathbf{B}'\mathbf{D}_c^{-1}\mathbf{B} = \mathbf{I}$.

There is a trivial part of the generalized SVD of $\mathbf{P}$ consisting of a singular value of 1 and the associated left and right singular vectors which is discarded before any results are displayed. The remaining left and right singular vectors define the orthogonal principal axes of the column points and row points respectively. In practice, the generalized SVD is computed indirectly by performing an ordinary SVD, where the ordinary SVD of any matrix $\mathbf{Q}$ is given by

$$\mathbf{Q} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}'$$

under the constraint $\mathbf{U}'\mathbf{U} = \mathbf{V}'\mathbf{V} = \mathbf{I}$. Thus, to compute the generalized SVD of $\mathbf{P}$, we perform the following steps:

i. Let $\mathbf{Q} = \mathbf{D}_r^{-1/2}\mathbf{P}\mathbf{D}_c^{-1/2}$.

ii. Obtain the ordinary SVD of $\mathbf{Q}$, giving $\mathbf{Q} = \mathbf{U}\mathbf{D}_\alpha\mathbf{V}'$.

iii. Let $\mathbf{A} = \mathbf{D}_r^{1/2}\mathbf{U}$, $\mathbf{B} = \mathbf{D}_c^{1/2}\mathbf{V}$, and $\mathbf{D}_u = \mathbf{D}_\alpha$.

iv. Then $\mathbf{P} = \mathbf{A}\mathbf{D}_u\mathbf{B}'$ is the required generalized SVD.

Table 1 Science Doctorates in the USA, 1960-1975

Discipline/Year	1960	1965	1970	1971	1972	1973	1974	1975
Engineering	794	2073	3432	3495	3475	3338	3144	2959
Mathematics	291	685	1222	1236	1281	1222	1196	1149
Physics	530	1046	1655	1740	1635	1590	1334	1293
Chemistry	1078	1444	2234	2204	2011	1849	1792	1762
Earth Sciences	253	375	511	550	580	577	570	556
Biology	1245	1963	3360	3633	3580	3636	3473	3498
Agriculture	414	576	803	900	855	853	830	904
Psychology	772	954	1888	2116	2262	2444	2587	2749
Sociology	162	239	504	583	638	599	645	680
Economics	341	538	826	791	863	907	833	867
Anthropology	69	82	217	240	260	324	381	385
Others	314	502	1079	1392	1500	1609	1531	1550

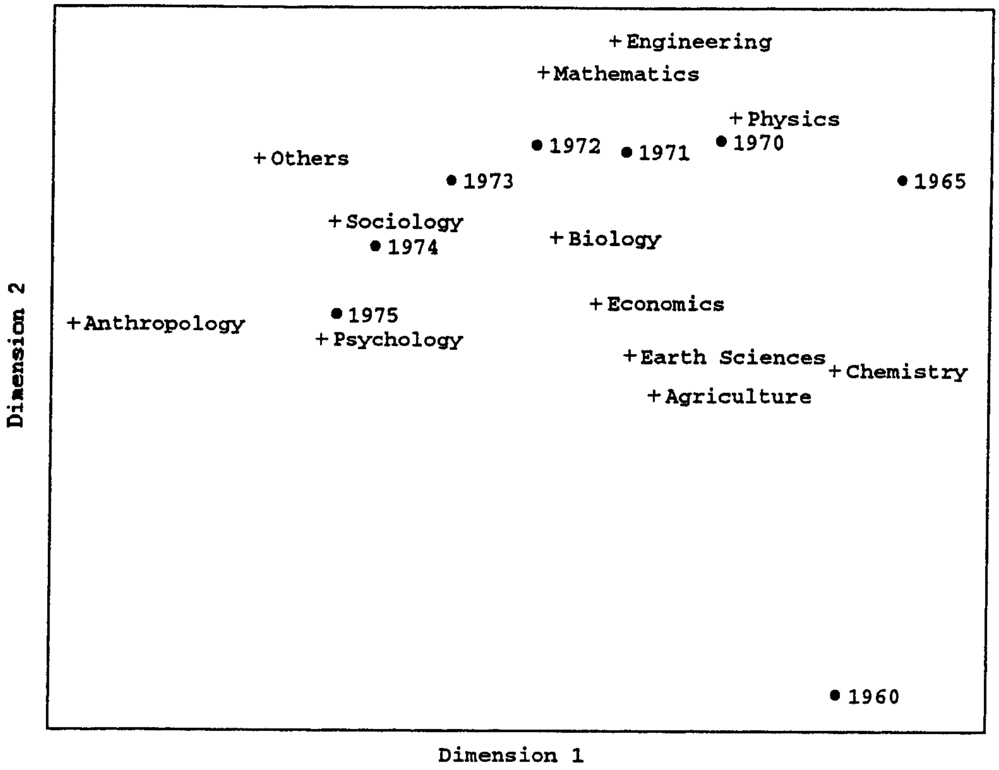


Figure 1 Correspondence Analysis of Science Doctorates Data

The row coordinates **F** and column coordinates **G** are then computed according to the appropriate selection of the formulas in Table 2. Greenacre (1984) introduced the terms "principal" and "standard" coordinates to distinguish between the two most common normalizations in literature. Standard coordinates are the coordinates $\mathbf{D}_r^{-1}\mathbf{A}$ or $\mathbf{D}_c^{-1}\mathbf{B}$ having unit normalization, while principal coordinates are the coordinates $\mathbf{D}_r^{-1}\mathbf{A}\mathbf{D}_u$ or $\mathbf{D}_{c-1}\mathbf{B}\mathbf{D}_u$ having weighted sums of squares equal to the squared singular values:

$$\mathbf{F}'\mathbf{D}_r\mathbf{F} = \mathbf{D}_u^2 \qquad \mathbf{G}'\mathbf{D}_c\mathbf{G} = \mathbf{D}_u^2.$$

The joint plot of the rows and columns in k dimensions, where $k \le \min(n-1, m-1)$, is obtained from the first k columns of the matrices **F** and **G**. A symmetric plot displays both the row points and column points in principal coordinates, whereas an asymmetric plot displays one set of points in principal coordinates and the other set of points in standard coordinates.

The squared singular values, or "principal inertias", quantify the amount of variation accounted for by the corresponding principal axes. If a large percentage of the total inertia lies along the k principal axes, it means that the chi-square distances among row profiles and among column profiles are well represented along these axes. Note that in an asymmetric plot, the principal inertias refer only to the set of points displayed in principal coordinates.

Table 2 Formulas for Coordinates

Analysis Options	Row Coordinates	Column Coordinates
Analyze Row Profiles	$\mathbf{F} = \mathbf{D}_r^{-1}\mathbf{A}\mathbf{D}_u$	$\mathbf{G} = \mathbf{D}_c^{-1}\mathbf{B}$
Analyze Column Profiles	$\mathbf{F} = \mathbf{D}_r^{-1}\mathbf{A}$	$\mathbf{G} = \mathbf{D}_c^{-1}\mathbf{B}\mathbf{D}_u$
Analyze Both	$\mathbf{F} = \mathbf{D}_r^{-1}\mathbf{A}\mathbf{D}_u$	$\mathbf{G} = \mathbf{D}_c^{-1}\mathbf{B}\mathbf{D}_u$

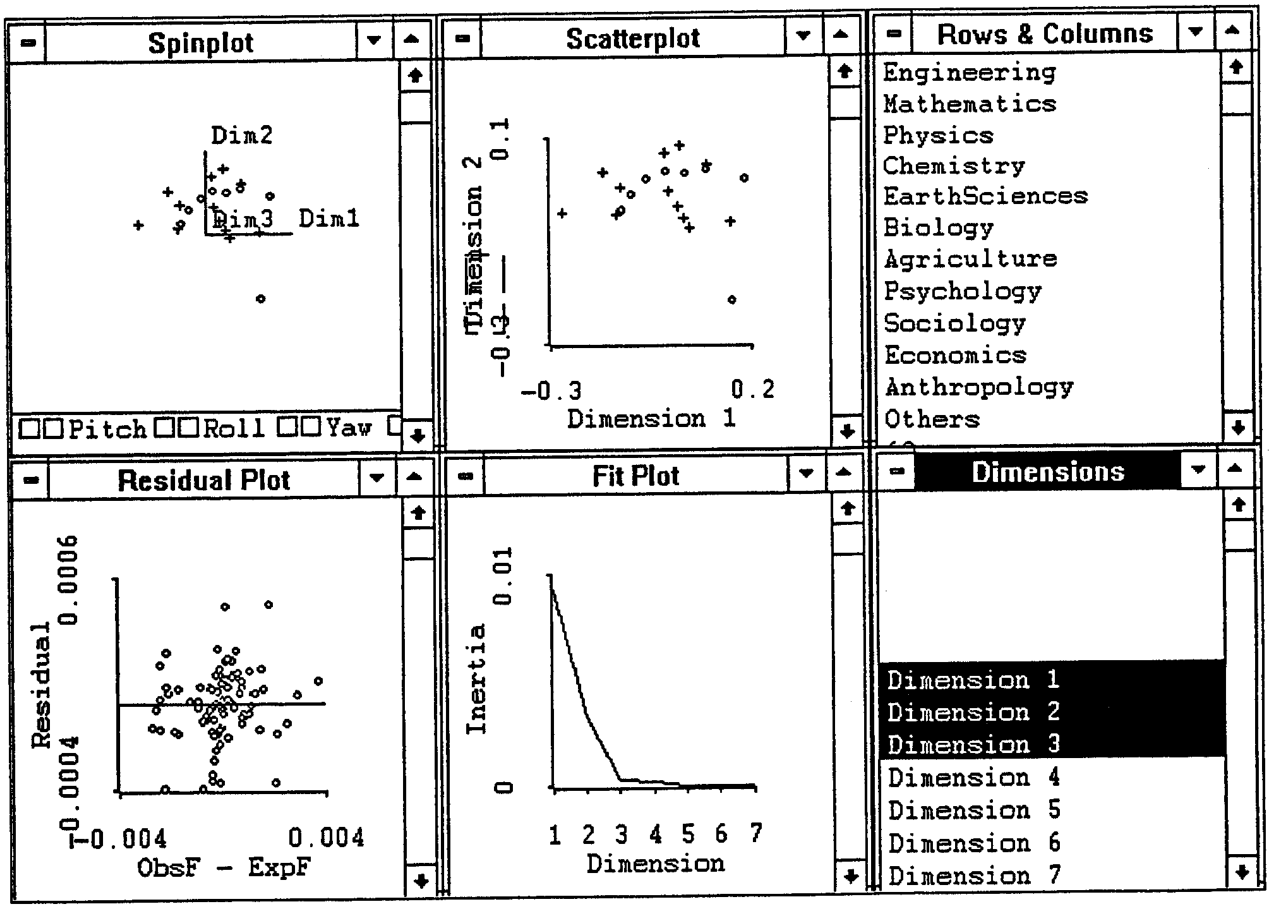
3 Statistical Visualization

The statistical visualization of correspondence analysis in ViSta-CA presents the results of the analysis in a group of interacting plots, called spreadplot (Young, 1994), which is based on the notion of a "graphical spreadsheet". The individual plots can be thought of as "cells" in the spreadsplot that can communicate with other cells via equations that define their relationships. Figure 2 shows the spreadplot for correspondence analysis of the Science Doctorates data.

The **Spinplot** is a plot of the row and column points in the first three of the dimensions selected in the **Dimensions** window. The mouse can be in one of three modes: *Spinning*, *Brushing*, and *Selecting*. The default mouse mode is *Spinning*. In this mode, the cursor looks like a hand. Holding the mouse button down and moving the cursor around the plot causes the plot to rotate. If you first hold the shift key down, then the plot will continue to rotate when you let up on the mouse button. You can also make the plot rotate by using the `Pitch`, `Roll`, and `Yaw` buttons at the bottom. When you place the mouse mode in *Brushing*, the cursor looks like a tiny paint brush with a rectangle attached to it. Moving the brush across the plot selects the points in the rectangle and highlights these points. When the mouse mode is changed to *Selecting*, the cursor looks like an arrow and any points that are clicked on will be selected and highlighted. In addition, if the cursor is dragged across an area, any points inside the area are also selected and highlighted. Labels of selected points will be shown in whatever plots are linked to the **Spinplot**. With the **Spinplot**, the analyst can search for those views in the various three-dimensional perspectives that display to him interesting structure of the geometric model.

The **Scatterplot** plots the first two dimensions that are selected in the **Dimensions** window. This plot has two mouse modes—*Brushing* and *Selecting*—which are the same as those modes for the **Spinplot**. The information in the **Scatterplot** was displayed in Figure 1.

Figure 2 Visualization Spreadplot for Correspondence Analysis



The **Rows & Columns** window, which contains the labels for the row and column points, is useful for locating or identifying points in the **Spinplot**, **Scatterplot**, and **Residual Plot** windows. Since each cell frequency corresponds to the intersection of a row and a column in a contingency table, when more than two labels are selected or when the two labels belong to the same way of the table, the points in the **Residual Plot** window will not respond to the selection.

The **Residual Plot** is a plot of the residuals versus the centered observed frequencies. The centered data are calculated by the formula $\mathbf{P} - \mathbf{rc}'$. The reconstitution of the correspondence matrix $\mathbf{P}$ based on the rank k weighted least squares approximation is given by the formula

$$\hat{\mathbf{P}} = \mathbf{A}_{[k]}\mathbf{D}_{u[k]}\mathbf{B}'_{[k]}$$

where the subscripts $[k]$ refer to the fact that only k of the dimensions are involved in the calculation. The specific columns of $\mathbf{A}_{[k]}$ and $\mathbf{B}_{[k]}$ that are involved correspond to the dimensions selected in the **Dimensions** window, which are not necessarily the first k singular vectors. The specific diagonal elements of $\mathbf{D}_{u[k]}$ are the associated singular values. The residual matrix is given by

$$(\mathbf{P} - \mathbf{rc}') - \hat{\mathbf{P}}.$$

The **Residual Plot** can be used for diagnostic checking as in a regression analysis.

The **Fit Plot** is a plot of the principal inertias against each dimension, showing the relative amount of fit for each dimension of the analysis. It serves the same purpose as the scree plot in principal component analysis.

The **Dimensions** window contains a list of dimensions. It serves as a control panel for the visualizations in the **Spinplot**, **Scatterplot**, and **Residual Plot** windows. Selecting at least two dimensions will change the current display of the row and column points in the **Scatterplot** window to that formed by the first two selected dimensions. For example, shift-clicking `Dimension 2`, `Dimension 3`, and `Dimension 5` produces a display of the points in the second and third dimensions. Selecting three or more dimensions will change the display in both the **Spinplot** and **Scatterplot**. In addition, selections in the **Dimensions** window are tantamount to a re-specification of the dimensionality of analysis—the k selected dimensions will determine the k singular vectors from the matrices $\mathbf{A}$ and $\mathbf{B}$ and the associated singular values from the diagonal matrix $\mathbf{D}_u$ that are to be used to calculate $\hat{\mathbf{P}}$ and the associated residuals. The residuals will be updated and re-plotted in the **Residual Plot** window to reflect the change in fit.

When the visualizations provided by the spreadplot is combined with the traditional reporting technique, which is also available in ViSta-CA, the analyst gains a greater understanding of the results of correspondence analysis than when either technique is used alone.

4 Statistical Re-Vision

Statistical re-vision is a set of statistical visualization tools that is used to help the analyst search for meaningful and parsimonious model parameterizations. In ViSta-CA, the analyst is able to move row or column points to new locations which may be more "interpretable", but which no longer satisfy all of the geometric properties of correspondence analysis.

When the analyst moves a point, the software responds by adjusting the positions of the other points so that they approximate the correspondence analysis equations as well as possible. For example, when a column point is moved by the analyst, the software calculates new positions for the row points.

The calculations of the new positions of the "other" set of points is done so that the basic relationship $\mathbf{P} = \mathbf{AD}_u\mathbf{B}'$ is maintained. This is done by noting that

$$\mathbf{P} = \mathbf{AD}_u\mathbf{B}' = \mathbf{D}_r\mathbf{FG}'\mathbf{D}_c$$

when the normalization is asymmetric (Analyze Row Profiles or Analyze Column Profiles). However, the relationship specified by the equation does not hold when the normalization is symmetric (Analyze Both), which is why point-moving is not possible in that case.

Understanding of the statistical re-vision technique may be enhanced through the use of examples. To this end, consider the spreadplot for the correspondence analysis of the Science Doctorates data shown in Figure 3. In the **Spinplot** and **Scatterplot** windows, the column points are displayed in principal coordinates; the row points, which are represented in standard coordinates, are masked using the **Hide Row Points** menu item in the **Scatterplot** menu. When normalization is asymmetric, the **Scatterplot** window supports an additional mouse mode—*Point-Moving*. In this mode the cursor looks like a finger, with which the analyst can move a column point by clicking on the point and dragging it to a new location.

The two-dimensional correspondence plot in the **Scatterplot** window, which accounts for approximately 95% of the total inertia, is almost an exact display of the column profiles. The spread of the column points along the first axis, Dimension 1, indicates a deterministic trend; whereas the second axis, Dimension 2, is difficult to interpret. To facilitate interpretation, the analyst may decide to move the 1960 year point in a way such that the distance between 1960 and 1965 is approximately equal to the distance between 1965 and 1970, to reflect the five-year gap (note that the other points are separated by a one-year interval).

When the year point 1975 is moved, the column coordinates $\mathbf{G}$ is changed to, say, $\tilde{\mathbf{G}}$. We must calculate a new set of row coordinates $\tilde{\mathbf{F}}$ such that

$$\mathbf{P} = \mathbf{D}_r \tilde{\mathbf{F}} \tilde{\mathbf{G}}' \mathbf{D}_c .$$

Note that $\tilde{\mathbf{F}}\tilde{\mathbf{G}}' = \mathbf{D}_r^{-1} \mathbf{P} \mathbf{D}_c^{-1}$.

We solve for $\tilde{\mathbf{F}}$ by the equation

$$\tilde{\mathbf{F}} = \mathbf{D}_r^{-1} \mathbf{P} \mathbf{D}_c^{-1} \left[\tilde{\mathbf{G}} \left(\tilde{\mathbf{G}}' \tilde{\mathbf{G}} \right)^{-1} \right].$$

While the basic relation $\mathbf{P} = \mathbf{D}_r \tilde{\mathbf{F}} \tilde{\mathbf{G}}' \mathbf{D}_c$ is maintained, the orthogonality constraint of correspondence analysis may be violated since the left singular vectors are related to the row coordinates through the equation $\mathbf{A} = \mathbf{D}_r^{-1} \mathbf{F}$. The "principal axes" defined by the new set of "left singular vectors"

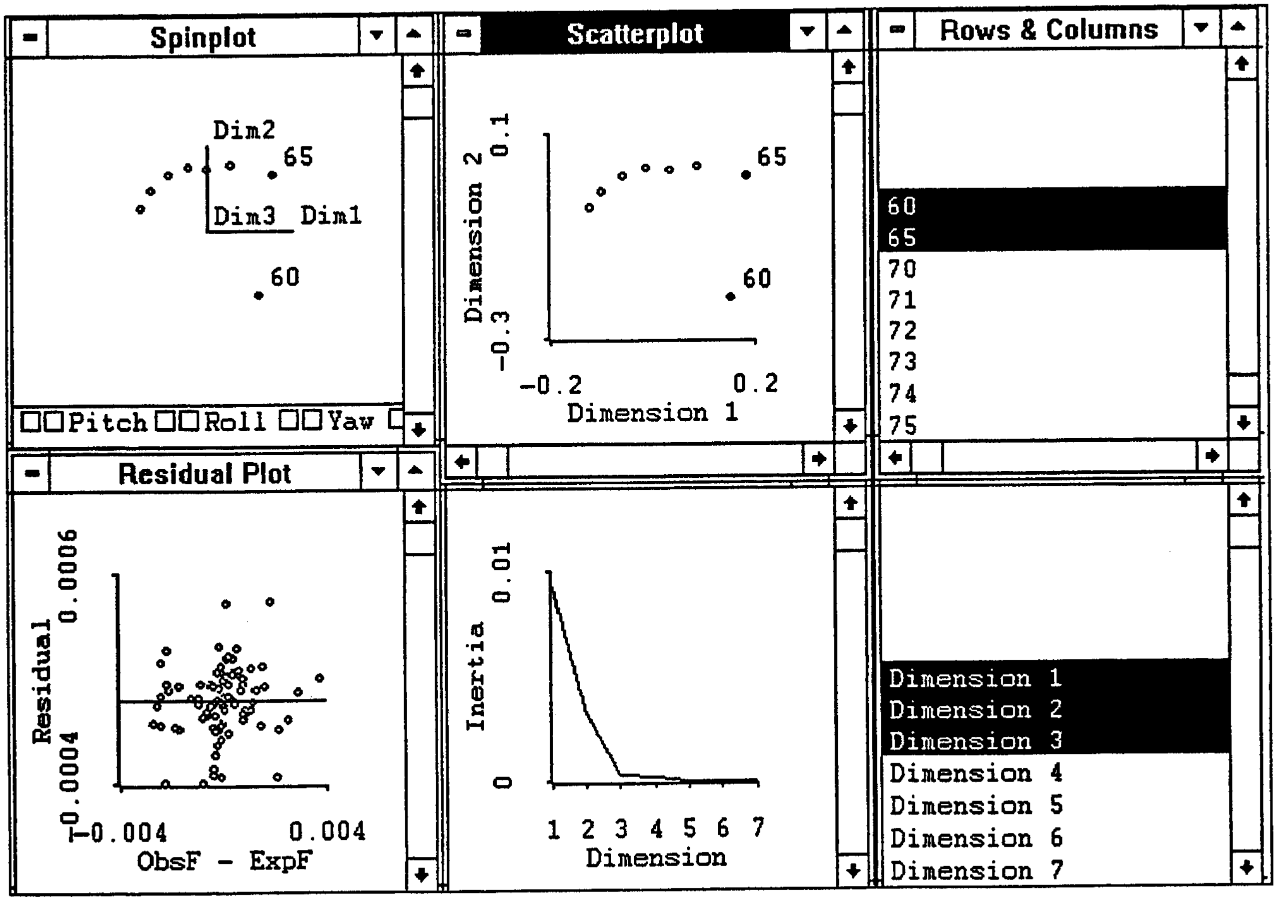
$$\tilde{\mathbf{A}} = \mathbf{D}_r^{-1} \tilde{\mathbf{F}}$$

may no longer satisfy the orthogonality constraint

$$\tilde{\mathbf{A}}' \mathbf{D}_r^{-1} \tilde{\mathbf{A}} = \mathbf{I}.$$

The new row coordinates are displayed in both the **Spinplot** and **Scatterplot** if the row points are not masked.

Figure 3 Correspondence Analysis of Science Doctorates Data—Asymmetric Normalization With Column Points In Principal Coordinates.



The residuals are re-calculated using the new values in $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$ to update the **Residual Plot**. To obtain an approximate measure of the quality of fit after point moving, we calculate a new set of "inertias" by the equation

$$\tilde{\mathbf{D}}_u = \left(\tilde{\mathbf{A}}'\tilde{\mathbf{A}}\right)^{-1}\tilde{\mathbf{A}}'\mathbf{D}_r\tilde{\mathbf{F}}$$

and plot the squared diagonal entries against each dimension as a dashed line in the **Fit Plot** window. Note that since the orthogonality constraint has been violated, the squared diagonal entries of $\tilde{\mathbf{D}}_u$ will overestimate the true inertias.

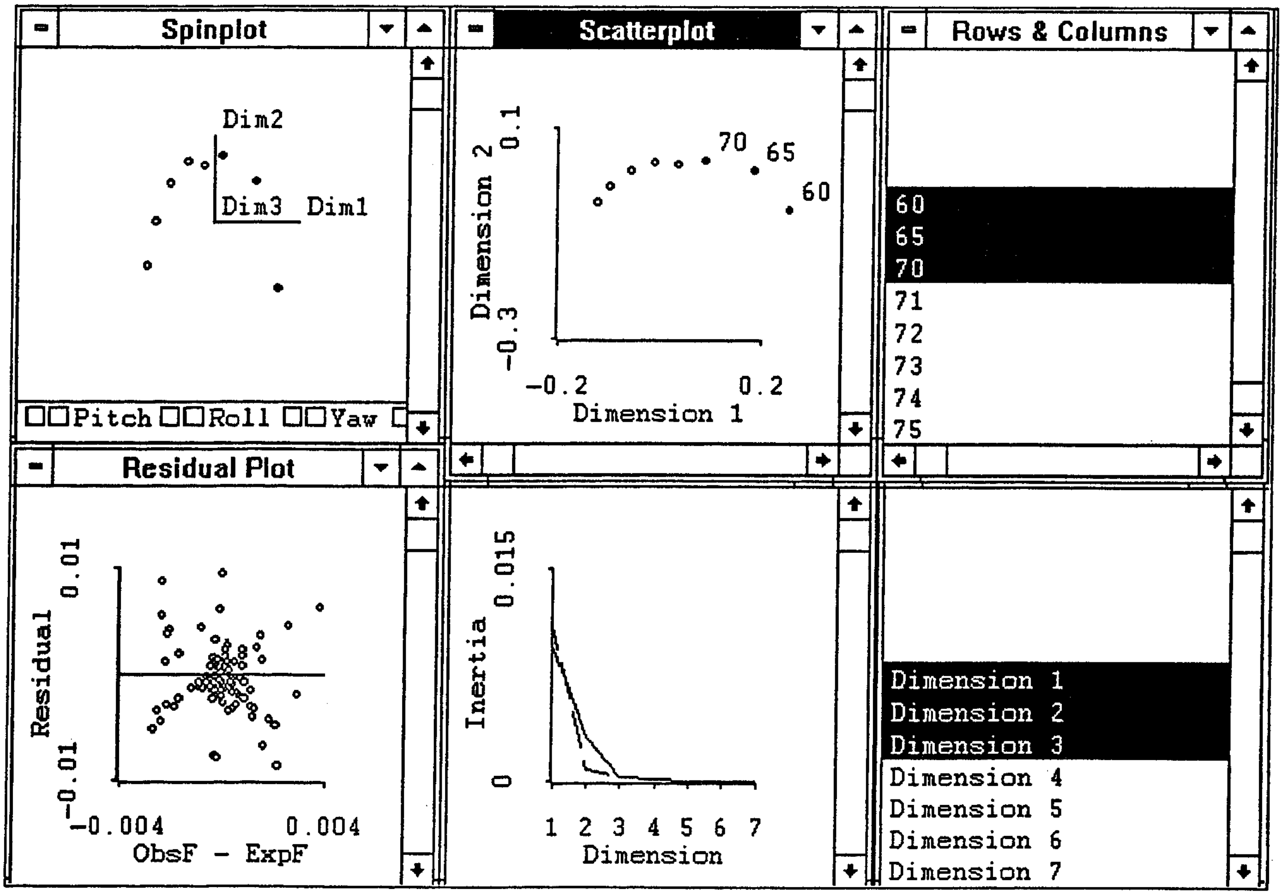
The results of moving the 1960 column point is presented visually in Figure 4. Notice that in the **Fit Plot** window, the inertia along the second axis decreased, reflecting the fact that the variation of the column points in the second dimension has been reduced. In addition, the magnitude of the residuals in the **Residual Plot** has increased.

5 Conclusion

ViSta-CA is a widely applicable tool for research involving correspondence analysis. It features state-of-the-art statistical visualization techniques for exploring the structure of the geometric model. When this technique is combined with the traditional reporting techniques, the analyst may gain considerable insight into the multidimensional properties of his data. A key feature of ViSta-CA is statistical revision, which allows the analyst to explore for a model that provides a better interpretation of the data than the one provided by traditional algebraic analysis. The principle behind this design is best summarized by a quotation from Marriott (1974):

> If the results disagree with informed opinion, do not admit a simple logical interpretation and do not show up clearly in a graphical presentation, they are probably wrong. There is no magic about numerical methods...

Figure 4 Statistical Re-Vision For Correspondence Analysis



6 References

1. Carroll, J.D., Green, P.E., and Schaffer, C.M. (1986) *Interpoint Distance Comparisons in Correspondence Analysis,* Journal of Marketing Research, Vol 23 (August), 271-280.
2. Carroll, J.D., Green, P.E., and Schaffer, C.M. (1989) *Reply to Greenacre's Commentary on the Carroll-Green-Schaffer Scaling of Two-Way Correspondence Analysis Solutions*, Journal of Marketing Research, Vol 26 (August), 366-368.
3. Greenacre, M.J. (1984) *Theory and Applications of Correspondence Analysis*, Academic Press, London.
4. Greenarce, M.J. (1989) *The Carroll-Green-Schaffer Scaling in Correspondence Analysis: A Theoretical and Empirical Appraisal*, Journal of Marketing Research, Vol 26 (August), 358-365.
5. Hoffman, D.L. and Franke G.R. (1986) *Correspondence Analysis: Graphical Representation of Categorical Data in Marketing Research*, Journal of Marketing Research, Vol 23 (August), 213-227.
6. SAS Institute, Inc. (1989) *The CORRESP Procedure.* In: SAS/STAT ® User's Guide, Version 6, 4th Edition, Vol 1, 615-676. Cary, NC, SAS Institute Inc.
7. Tierney, L. (1990) *Lisp-Stat: An Object-Oriented Environment for Statistical Computing and Dynamic Graphics*, John Wiley, New York.
8. Young, F.W. (1994) *Vista - The Visual Statistics System. Overview and Tutorial.* Psychometric Laboratory Memorandum 94-I, University of North Carolina, Psychometric Laboratory.

Visualized Models for Binary Response Data

Christopher A. Wiesen
Psychometric Laboratory
University of North Carolina at Chapel Hill
Davie Hall, CB 3270
Chapel Hill, NC. 27599-3270

Abstract

Logistic regression is the accepted parametric method for analyzing data with continuous predictors and a binary response. As with general linear models the relation between the predictors and the logit of the response probability is assumed linear. When the observed response is continuous, visual techniques, such as scatterplots, are useful in ascertaining the nature of this relation, but scatterplots offer little information when the observed response is binary. A system offering visual model exploration techniques, derived specifically for binary response data, is proposed.

1.0 Introduction

Many models used by social and biological statisticians fall within the realm of generalized linear models. These models consist of three components. The random component, comprising the independent observations of the dependent variable y, the systematic component which is the explanatory model $\theta = \sum ß_i x_i$ where i indexes the independent variables, and the link function $f(E(y)) = \theta$. The simplest link is the identity link, $f(E(y)) = E(y)$.

Correct application of generalized linear models includes the assumptions of linearity and additivity. The linearity assumption specifies that a straight line describes the relation between x_i and θ, that is a unit change in x_i always yields the same change in θ. Additivity means that there are no interactions; a change in any x_i results in the same change in θ independent of the values of all x_j $i \neq j$.

Violation of these assumptions will lead to model misspecification. Incorrectly forcing a linear additive fit where the association is nonlinear or nonadditive results in incorrectly large error and systematically biased predicted values. For example, if the true relation is quadratic, the linear model may result in $ß_i = 0$, indicating that there is no association between x_i and y.

The methods for applying the generalized linear models to continuous response variables and one or more continuous or categorical predictor variables are well understood, especially linear models solved by the least squares normal equations. The solution to these equations yields the unbiased estimates for the equation $E(y) = \alpha + \sum ß_i x_i$.

2.0 Binary Responses

A binary response presents problems for the general linear model. Because the response is categorical, the normal equations will not yield reasonable solutions for the regression of y on x. Since the analyst will likely be more interested in the probability of a response conditional on having observed x than the specific value of the outcome, we consider $p(x) = p(y = 1|x)$, the probability of responding 1 given x, as the response variable of interest. Clearly, p(x) is not suitable for use as a linear model response variable as it is bounded by 0 and 1. A linking function is required to transform p(x) to a variable that is continuous, unbounded and may reasonably be expected to have linear relation with x_i. Logistic regression employs the logit link, that is $\theta = \log(p(x_i)/(1 - p(x_i)))$.

3.0 Visualizing GLM's

If one is unsure of the shape of the relation between y and x_i

one may choose to construct a scatterplot. This allows visual inspection of the relation and may suggest that a transformation of the x_i variable is necessary to effect a linear model. That is the model $E(y) = x_i + x_i^2$ or $E(y) = \log(x_i)$ may better describe the linear model than would $E(y) = x_i$.

3.1 Smoothing

Another choice for the relation between x_i and y is the function $E(y) = S(x_i)$ where S is a nonparametric function, such as a smoother. Here the predicted value of y is found by a weighted average of the y's for observations that lie in close proximity (in the x space) to the target observation. Weighting schemes include simple means as well as linear and polynomial smoothers (Cleveland, 1979; Cleveland, Devlin and Grosse, 1988; Cleveland and Devlin, 1988; Fan, 1992).

As in the continuous response case, the analyst may wish to visualize binary response data in conjunction with analysis. Unfortunately, a scatterplot is of little utility when the response variable is binary as the plot will merely be rows of points at 0 and 1. A solution to this problem is found in smoothing (Copas, 1983), where the smoothed response variable is $p(x_i)$. Here smoothing is used not necessarily to form a model but rather to visualize the shape of $p(x_i)$ vs x_i.

If the logistic regression model appears to fit the smoothed $p(x_i)$ then we may choose that model. If not then we may wish to transform the x_i variable. Transformations of an x_i variable may not be immediately suggested by the shape of $S(x_i)$. Since logistic regression assumes a linear relation between x_i and $\text{logit}(p(x_i))$, observing the plot of $\text{logit}(S(x_i))$ vs x_i may prove useful. This plot may be used to choose some transformation of x_i, in much the same way a scatterplot is used with a continuous response variable. As in the continuous outcome case, the model may also be defined by a smooth.

4.0 Visualization For Binary Response Data in the XLisp-Stat Environment

XLisp-Stat provides an ideal environment for implementing the ideas discussed for visualizing models with continuous predictors and binary responses. Smoothing is a computationally intensive procedure that requires visualization for a true appreciation and understanding of the result. XLisp-Stat offers both the computational efficiency and high resolution graphics to effectively smooth binary response models.

The proposed system provides two stages of data analysis. At stage 1 the user smoothes the data using generalized additive model methods (Hastie and Tibshirani, 1989). The resulting smooth is then inspected visually. Visual techniques include:

1) Plots of both the smoothed probability and smoothed logistic surface with predicted values.

2) Residual plots.

3) The marginal smooth for each independent variable.

Statistics indicating the importance of each variable in the model are also given.

After inspecting the smooth, the user may go to stage 2, fitting a parametric model. The visual parametric techniques include:

1) Biplots with a vector indicating the relative magnitude of the effect of each variable in the model.

2) The predicted response surface with predicted values.

3) Residual plots for each independent variable.

4) Influence plots.

Parameter estimates and standard errors are also included.

The user may, at any time, alter the **X** matrix by adding or dropping variables or transforming variables. The result of adding a transformed variable will be seen in the predicted response surface.

5.0 Example

Figure 1 shows the smooth for data generated by the model $\text{logit}(y) = 40{*}x_1 + 0{*}x_2 + 40{*}x_3 + 0{*}x_4$. The upper left plot is the function p = inverse logit $[S(x_1) + S(x_4)]$; the upper middle plot is the function $y = S(x_1) + S(x_2)$. Both plots include predicted values for the full model for all observations. At the lower middle is the residuals plot and at the lower left is the single dimension plot of x_4. All of these plots are dynamic as the variables viewed are changeable. The upper right window contains observation names while the lower right window is for statistics. The statistics SSQ and %Total indicate the contribution of each independent variable to the overall variance of the predicted logits, but assume that the **X** matrix is orthogonal.

Inspecting the various plots and windows indicates that:

1) The smooth adequately describes the data.

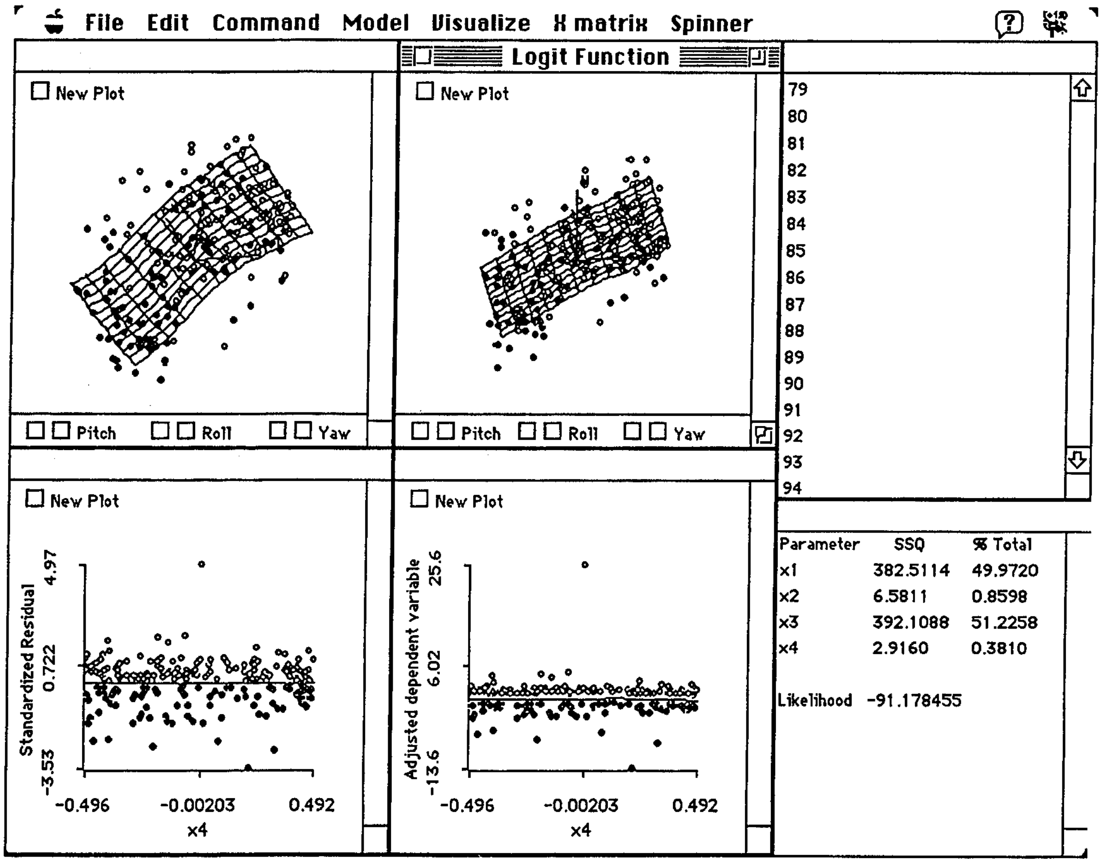
2) The relation between each x_i and the response logit(y) is linear.

3) The variables x1 and x3 are salient while x2 and x4 are not.

The **X matrix** menu option may be used to remove x_2 and x_4 from the **X** matrix and a parametric model is fitted using the Model menu option **Parametric**. The resulting model is shown in Figure 2. The plots are, clockwise from upper left, a biplot of independent variables with the parameter vector added, a probability function plot with full model predicted values, an influence plot and a residuals plot. All plots are dynamic in that the variables viewed may be changed. The observation window is as in the smooth model and the statistics window contains statistics common for a logistic regression analysis.

The point indicated by a "+" is an observed 1 that had a predicted probability near 0; it has both a large residual (lower left) and a large effect on the chi-square (lower right). Had this been actual data, thsese findings could indicate that a closer inspection of this observation was necessary.

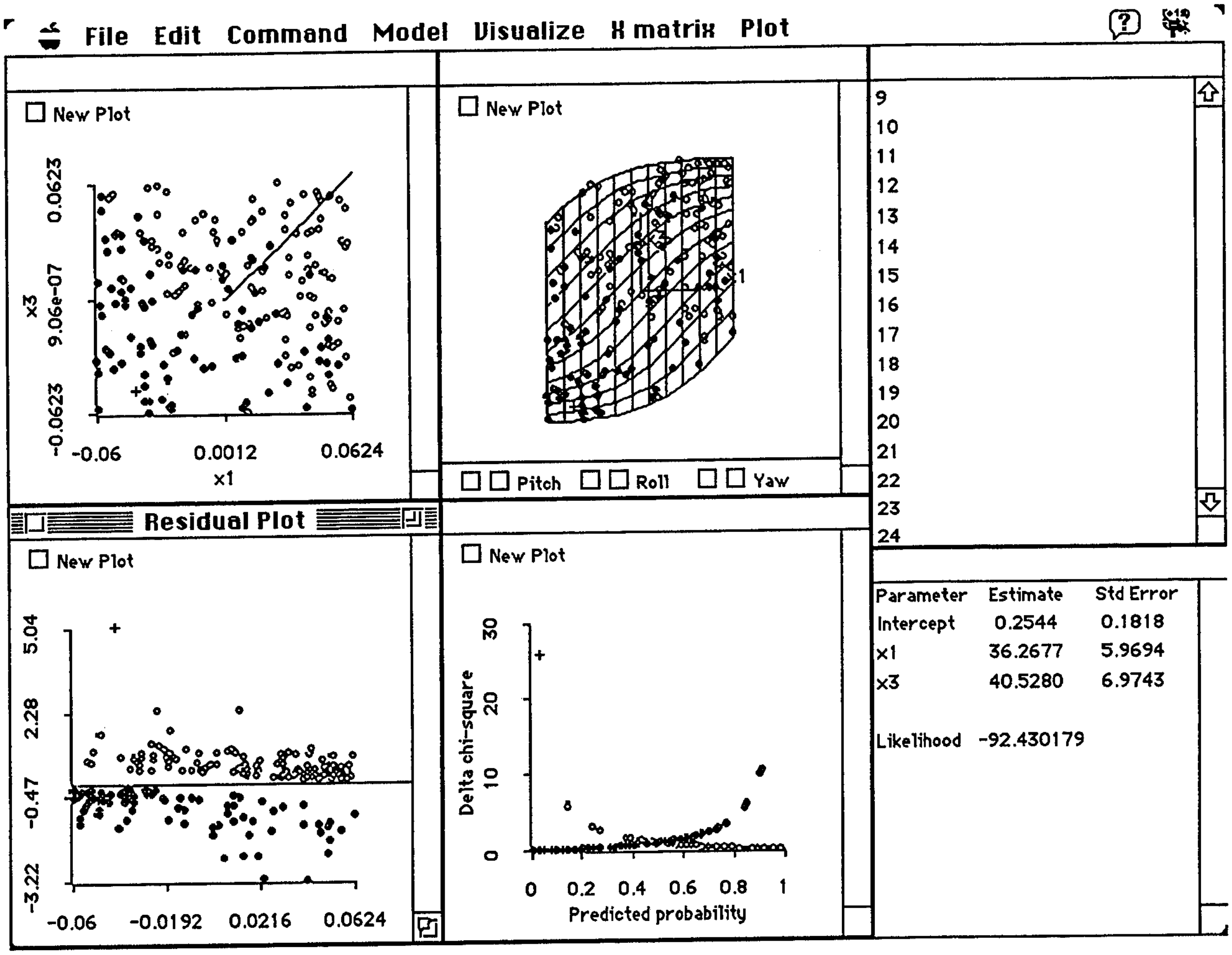
Figure 1. A smooth model



Parameter	SSQ	% Total
x1	382.5114	49.9720
x2	6.5811	0.8598
x3	392.1088	51.2258
x4	2.9160	0.3810

Likelihood -91.178455

Figure 2. A parametric model



```
Parameter   Estimate   Std Error
Intercept   0.2544     0.1818
x1          36.2677    5.9694
x3          40.5280    6.9743

Likelihood  -92.430179
```

6.0 References

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association, 74*, 829-836.

Cleveland, W. S., and Devlin, S. J. (1988). Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the America Statistical Association, 83*, 596-610.

Cleveland, W. S., Devlin, S. J. and Grosse, E. (1988). Regression by local fitting: methods, properties, and computational algorithms. *Journal of Econometrics, 37*, 87-144.

Copas, J. B. (1983). Plotting *p* against *x*. *Applied Statistics, 32*, 25-31.

Fan, J. (1992). Design-adaptive nonparametric regression. *Journal of the American Statistical Society, 87*, 998-1004.

Hastie, T. and Tibshirani, R. (1989). Generalized additive models. *Statistical Science, 1*, 297-318.

Exploring High-Dimensional Data with Visual Components Analysis

Richard A. Faldowski
Psychometric Laboratory
University of North Carolina at Chapel Hill
Davie Hall, CB 3270
Chapel Hill, NC. 27599-3270

Abstract

Principal components analysis is a well known statistical model used to approximate a high dimensional data space by a subspace of lower dimensionality. Like many multivariate statistical procedures, when the principal components model is fit to data based on a purely algebraic criteria, it can be plagued by problems of data sensitivity and interpretability. An interactive technique called visual components analysis is proposed as one solution to these difficulties. Visual components analysis allows a user to visualize, evaluate, and modify a principal components model within a unified graphical environment. It is believed that visual components analysis will yield subjectively more satisfying solutions than solutions obtained from classical algebraic analyses.

1.0 Motivation

The principal component model is a well-known statistical model commonly used for reducing data dimensionality, assessing linear relationships in data, or identifying the "latent dimensions" presumed to underlie observed variables. It is written:

$$\mathbf{X} = \mathbf{U}(\Lambda\mathbf{V}') = \mathbf{U}\tilde{\mathbf{V}}' \qquad \text{(EQ 1)}$$

where

- **X** is a matrix of *n* observations measured on *m* variables,
- Columns of **U** are unit standardized components,
- Matrix Λ^2 is the diagonal matrix of eigenvalues, and
- Columns of $\tilde{\mathbf{V}}'$ are component coefficients (parameters of the principal component model) with crossproducts equal to the eigenvalues.

Traditionally, the principal component model is fit through algebraic equations that both reflect desired data characteristics, as well as possess appropriate analytic properties. However, a variety of common data conditions can lead awry a principal component model fit by algebraic methods. Examples of such conditions include: data matrix ill-conditioning, outliers, leverage points, and influential observations (Barnett & Lewis, 1984; Jolliffe, 1986; Belsley, 1991; Critchley, 1985; Radhakrishnan & Kshirsagar, 1981).

Algebraic solutions to these difficulties have previously been proposed. They include *detection-based strategies*—that is, find the problematic variables, observations, or model characteristics and eliminate them— as well as *robust*, *resistant*, and *local* fitting methods. Unfortunately, the robust/resistant estimators and local fitting methods with the most desirable properties frequently suffer limitations that they are computationally intensive, require iterative solutions, and contain arbitrary constants[1] that substantially affect the solutions they ultimately attain (Belsley, 1991; Cleveland, 1993; Cleveland, Grosse & Shyu, 1991; Huber, 1981).

2.0 Statistical Revision

I propose an dynamic, user-interactive approach called *statistical re-vision* as an additional solution to the problems suffered by many ordinary algebraic statistical modeling techniques. Statistical re-vision is a cyclic, iterative approach to model fitting that utilizes the analyst as an active element in the statistical estimation process. Although it begins with algebraically optimal model parameter estimates, during the course of statistical re-vision operations, a subjective, aesthetic estimated parameter optimality is substituted for the initial algebraic criteria. Specific characteristics of the sub-

[1] E.g. tuning constants or bandwidths.

jective criteria are determined by the analyst (Young, Faldowski, & McFarlane, 1993).

Statistical re-vision conducted in two phases. The first phase is *interactive graphical modeling*. In it, the user graphically modifies the estimated parameters of the model by moving a representation of the model in a computer display. A new set of subjectively adjusted parameter estimates (coefficients) and predicted values for the model are produced as a result. The second phase is *interactive graphical exploration*. Here, the analyst explores the implications of his subjective adjustment of estimated parameters in terms of fit, and he has option of further refining his choice of the subjective parameter estimates.

When statistical re-vision is applied in the context of the principal component model, I call the resulting modeling process *visual components analysis* and the resulting set of components, *visual components*. The interactive graphical modeling phase of visual components analysis consists of user modification of the initial component coefficients matrix $(\tilde{\mathbf{V}}')$, resulting in a set of subjective component coefficients $(\tilde{\mathbf{V}}^{*\prime})$. Based on the new set of coefficients, a corresponding set of subjective components $\mathbf{U}^*$ are calculated.

The interactive graphical exploration phase in visual components analysis consists of graphical exploration of the components and coefficients derived from the altered parameter estimates using structure and fit plots. It also entails consideration of alternative sets of coefficients different from the initial ones $(\tilde{\mathbf{V}}')$, but not as extreme as those specified during interactive graphical modeling $(\tilde{\mathbf{V}}^{*\prime})$. The interactive graphical exploration phase of statistical re-vision is highly dynamic with plots of component structure and fit indices continually updated throughout the exploration process.

Note that when statistical re-vision is used to adjust the algebraically optimal parameters estimated from a set of data, "subjective" fit increases, but objective fit virtually always decreases. In addition, it is often necessary to violate primary constraints of the model. For example, the principal components model provides the only decomposition of a data matrix that is orthogonal in both scores and coefficients. During visual components analysis, one of these properties must be sacrificed. Since characteristics of variable space were assumed of primary interest, in the remaining discussion the orthogonality of component scores was selected to be maintained. In other applications, compelling arguments might be made for choosing the alternative constraint.

3.0 A System for Visual Components Analysis

Figure 2 shows mock-ups of plots from a statistical graphics system designed to support visual components analysis. It contains two general types of plots:

- *Structure plots*, which are designed to show the structure of the data and model, and
- *Fit plots*, which are designed to help the analyst assess the degree to which a component model objectively fits the data.

Through the joint use of these plots during statistical re-vision, the analyst attempts to balance the subjective quality of the structure displayed in the structure plots against the objective quantification of fit relayed by the fit plots. The visual components system is designed to help the analyst balance trade-offs between subjective and objective fit as he attempts to optimize subjective characteristics of the components solution.

The structure plots include the two "BiPlot" and the "Tour Plot" windows, which present the structure of the data and model as classic biplots (Gabriel, 1972). The "TourPlot" also serves as a control center for the system. It manages which space (data space, model space, error space, or interactive-graphical-exploration space) is currently visible in the structure plots. It controls whether the system is operating in visualization or statistical re-vision modes. In addition, it supports guided tours (Young, Kent & Kuhfeld, 1988; Buja, Asimov, Hurley & McGill, 1988) between the spaces shown in the "BiPlot" windows and provides graphical tools for use in the two phases of statistical re-vision. The "BiPlot" windows, meanwhile, control what variables are displayed in the "TourPlot" and show the initial and target spaces for guided tours presented in the "TourPlot".

The "Scree Plot" is a standard display in principal components analysis. It portrays the variances of the components plotted against component number. The "Variable-Model Variance Trace Plot" shows what percent of each variable's variance is accounted for by the current model components. The "Variable-Component Variance Trace Plot" shows what percent of each variable's variance is accounted for by specific components within the current components model.

4.0 Interactive Graphical Modeling

Although interactive graphical modeling for visual components analysis may be performed in either the model or data spaces, for illustrative purposes I will describe it in the model space. To begin interactive graphical modeling, the analyst switches from visualization to re-vision mode in the "TourPlot". At that point, he gains access to the interactive graphical modeling tools shown at the bottom of the "TourPlot" window. The "Direct" and "Indirect" buttons describe two ways of performing interactive graphical modeling. Figure 1 shows the "TourPlot" window during "Direct" interactive graphical modeling mode. In the left-hand panel, note that the cursor has changed to a finger which the analyst used to "grab" one of the component vectors in the display. He then orthogonally rotated the component vector to a new location among stationary representations of the observations and variables. This is portrayed in the right-hand panel of Figure 1.

When the analyst finds a suitable new location for the component vectors, he presses the "Compute" button. At this point, the system translates the user's graphical rotation into an orthogonal transformation matrix, **R**, which is used to define a new set of adjusted coefficients and components, $\tilde{\mathbf{V}}'^*$ and $\mathbf{U}^*$, respectively. The components model, modified through interactive graphical modeling, may be written:

$$\mathbf{X} = (\mathbf{UR})(\mathbf{R}'\tilde{\mathbf{V}}') = (\mathbf{U}^*)(\tilde{\mathbf{V}}'^*) \qquad \textbf{(EQ 2)}$$

where

- **R** equals the orthogonal rotation matrix,
- $\mathbf{U}^*$ is the new graphically altered set of components, and
- $\tilde{\mathbf{V}}'^*$ is the new graphically altered set of coefficients (estimated model parameters).

The system now automatically enters the second phase of visual components analysis, interactive graphical exploration.

5.0 Interactive Graphical Exploration

As the system enters the interactive graphical exploration phase of visual components analysis, the information displayed in the structure plots ("BiPlot1", "BiPlot2", and "TourPlot" windows) change. Regardless of what space the interactive graphical modeling was performed in, during interactive graphical exploration, all structure plots show model spaces. The "BiPlot1" window displays the structure determined from the initial set of component coefficients, while the "BiPlot2" window displays the structure determined from the graphically-altered component coefficients. The "TourPlot", meanwhile, shows the structure of a set of components determined from a linear combination of the initial and the graphically-altered coefficients.

It is convenient to think about the structure shown in the "TourPlot" window during interactive graphical exploration as formed by conducting a guided-tour between the components represented in the "BiPlot1" window and the corresponding components in the "BiPlot2" window. Each step in the guided tour (a trigonometric interpolation between the model spaces shown in the "BiPlot1" and "BiPlot2" windows) defines an alternative composite set of components and parameter estimates. That is:

$$\mathbf{U}_i^{**} = (cos\theta_i)\,\mathbf{U} + (sin\theta_i)\,\mathbf{U}^* \qquad \textbf{(EQ 3)}$$

$$\mathbf{V}_i^{**} = (cos\theta_i)\,\mathbf{V} + (sin\theta_i)\,\mathbf{V}^* \qquad \textbf{(EQ 4)}$$

where

- $\mathbf{U}$ and $\mathbf{V}$ are the initial components and coefficients (defined prior to interactive graphical modeling),
- $\mathbf{U}^*$ and $\mathbf{V}^*$ are subjective components and coefficients (determined through interactive graphical modeling),
- $\mathbf{U}_i^{**}$ and $\mathbf{V}_i^{**}$ are the alternative, composite set of components and coefficients (determined at the ith step in a guided tour rotation during interactive graphical exploration), and
- θ_i is the cumulative rotation angle on the ith step, $[0° \le \theta_i \le 90°]$.

Note that each step in the guided tour results in a composite set of component coefficients different from the initial ones, but less extreme than those determined through interactive graphical modeling. In practice, it is also usually necessary to build an implicit correction factor into the guided tour rotation in order to maintain component orthogonality. This detail is a minor technicality that does not substantively alter the nature of the procedure.

The system is set up so that the analyst may rotate from the initial into the graphically altered components and back as many times as needed to fully appreciate the effects of the graphical alteration and to determine whether an intermediate set of coefficients is more appropriate or not. Throughout the interactive graphical exploration rotations, all of the fit displays are continually updated in order to give the analyst a sense of the objective quality of each intermediate set of parameter estimates. At any point, the analyst has the option of stopping the rotations and updating the initial components and coefficients with those from the currently visible composite set.

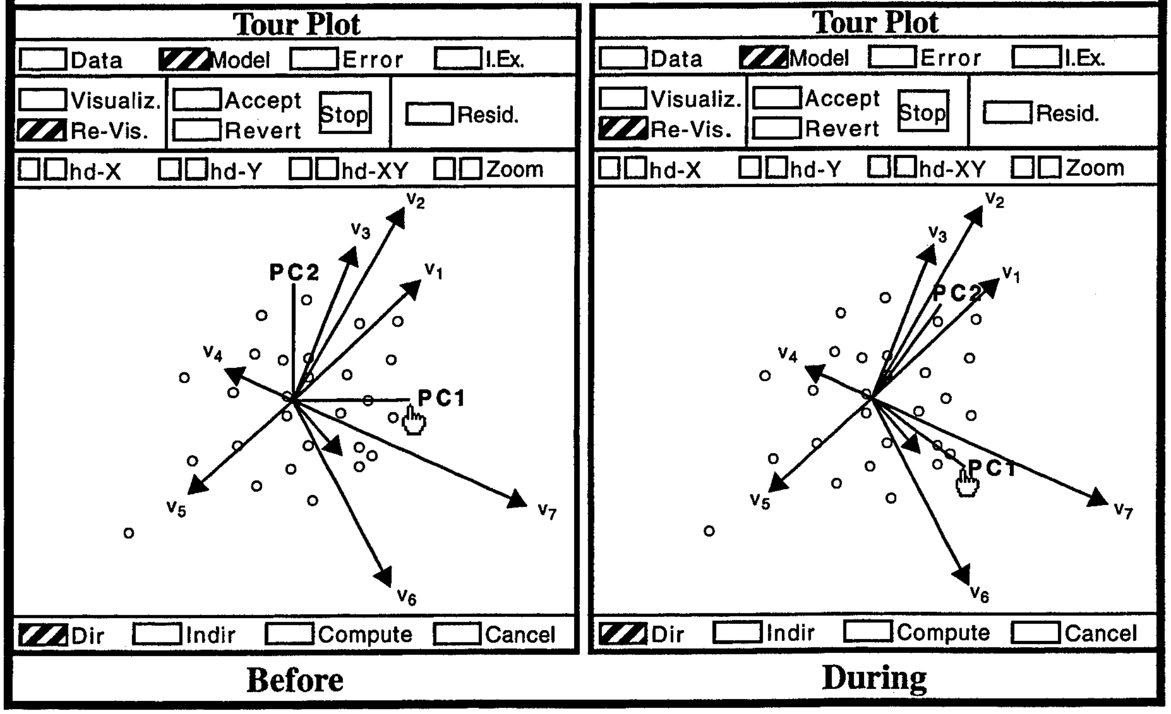
6.0 Conclusion

Statistical re-vision was presented as the framework within which visual components analysis was organized and it provided the structure through which visual component modeling interactions were carried out. Over a number of iterations through a cycle of visualization and statistical re-vision, it is anticipated that the analyst will generate visual components that mitigate many of the effects of outlying or influential observations in the component solution, that visual components should more closely conform with the analyst's knowledge about his substantive research problem, and that visual components analysis will yield a subjectively more satisfying solution than that obtained from classical algebraic component analyses.

7.0 References

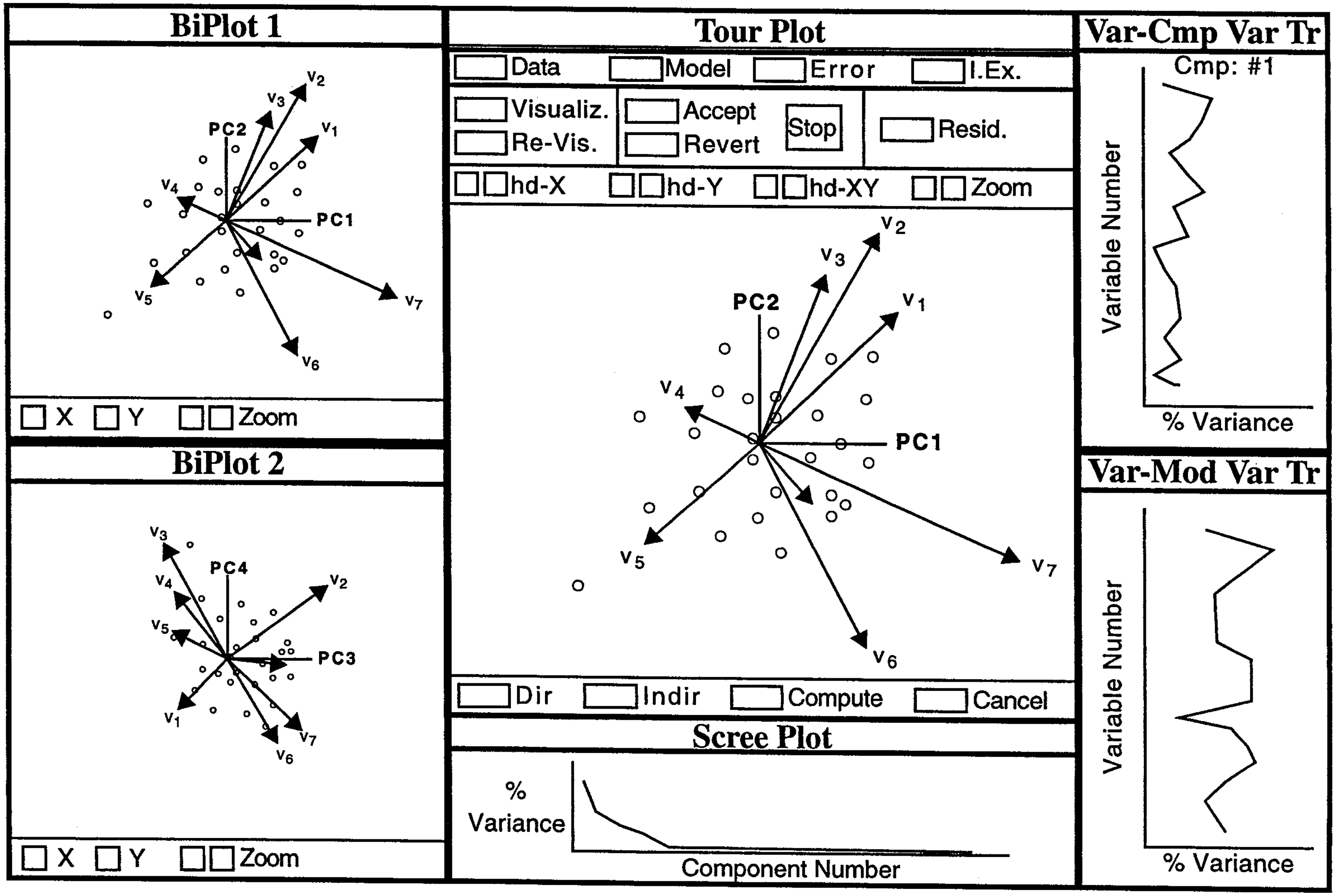
1. Barnett, V. & Lewis, T. (1984). *Outliers in Statistical Data* (2nd ed.). NY: Wiley.
2. Belsley, D.A. (1991). *Conditioning Diagnostics: Collinearity and Weak Data In Regression.* New York: Wiley.
3. Buja, A., Asimov, D., Hurley, C. & McDonald, J.A. (1988). Elements of a viewing pipeline for data analysis. In W.S. Cleveland & M.E. McGill (Eds.), *Dynamic Graphics for Statistics* (pp.277-308). Belmont, CA.: Wadsworth & Brook/Cole Advanced Books.
4. Cleveland, W.S. (1993). *Visualizing Data.* Murray Hill, N.J.: AT&T Bell Laboratories.
5. Cleveland, W.S., Grosse, E. & Shyu, W.M. (1991). Local regression models. In J.M. Chambers & T. Hastie (Eds.), *Statistical Models in S* (pp.309-376), NY.:Chapman and Hall.
6. Critchley, F. (1985). Influence in principal components analysis. *Biometrika*, **72**, 627–636.
7. Gabriel, K.R. (1971). The biplot-graphic display of matrices with application to principal components analysis. *Biometrika*, **58**, 453-467.
8. Huber, P.J. (1981). *Robust Statistics.* NY.: Wiley.
9. Jolliffe, I.T. (1986). *Principal Component Analysis.* New York: Springer-Verlag.
10. Radhakrishnan, R. & Kshirsagar, A.M. (1981). Influence functions for certain parameters in multivariate analysis. *Communications in Statistics—Theory and Methods*, **A10**, 515–529.
11. Young, F.W., Faldowski, R.A. & McFarlane, M.M. (1993). Multivariate statistical visualization. In C.R. Rao (Ed.), *Handbook of Statistics, Vol. 9* (pp.959-998). NY.: Elsevier Science Publishers.
12. Young, F.W., Kent, D.P. & Kuhfeld, W.F. (1988). Dynamic graphics for exploring multivariate data. In W.S. Cleveland & M.E. McGill (Eds.), *Dynamic Graphics for Statistics* (pp.391-424). Belmont, CA.: Wadsworth.

FIGURE 1. A View of Interactive Graphical Modeling in the Component Model Space.


Tour Plot
□Data ▨Model □Error □I.Ex.
□Visualiz. ▨Re-Vis. □Accept □Revert Stop □Resid.
□□hd-X □□hd-Y □□hd-XY □□Zoom
PC2
PC1
V1
V2
V3
V4
V5
V6
V7
▨Dir □Indir □Compute □Cancel
Tour Plot
□Data ▨Model □Error □I.Ex.
□Visualiz. ▨Re-Vis. □Accept □Revert Stop □Resid.
□□hd-X □□hd-Y □□hd-XY □□Zoom
PC2
PC1
V1
V2
V3
V4
V5
V6
V7
▨Dir □Indir □Compute □Cancel

Before **During**

FIGURE 2. A System for Performing Visual Components Analysis.



Monte Carlo Assessment of Influence and Sensitivity in Bayesian Modeling *

Mario Peruggia
Department of Statistics, The Ohio State University, Columbus, OH 43210-1247

Abstract

We propose *computationally feasible* diagnostics within the Bayesian paradigm, We focus on the detection of influential observations and the assessment of the sensitivity of the analysis to prior assumptions. We quantify differences in the inferential conclusions that might be drawn under modeling conditions that depart from an assumed setting by estimating, via a Monte Carlo approximation based on a single draw from the Gibbs Sampler, the Kullback-Leibler divergence of the baseline posterior distribution of the model parameters from the alternative posterior distributions obtained by deleting some observations or by altering the modeling assumptions. We illustrate these ideas in the context of a normal means hierarchical model.

1 Introduction

In this article we propose *computationally feasible* diagnostics within the Bayesian paradigm. We focus on two issues: (a) detection of influential observations, and (b) assessment of the sensitivity of the analysis to prior assumptions. In both cases we wish to quantify differences in the inferential conclusions that might be drawn under modeling conditions that depart from an assumed setting. We do so by measuring the Kullback-Leibler divergence (Kullback 1959) of the baseline posterior distribution of the model parameters from the alternative posterior distributions obtained by deleting some observations or by altering the modeling assumptions.

The difficulty with such an approach is that, in principle, it entails reperforming the analysis for each dataset/model considered. Within the Bayesian framework this implies repeated evaluations of multidimensional integrals to obtain the posterior distributions of the model parameters. While closed form analytic expressions for these posterior distributions are available for simple models (DeGroot 1986; Berger 1985), for more realistic cases either numerical quadrature methods (Smith et al. 1987), asymptotic approximations (Walker 1969; Tierney and Kadane 1986), or successive substitution sampling techniques (Gelfand and Smith 1990; Tanner 1991) must be used. The majority of these methods, with the exception of some asymptotic approximations, require a large computational effort.

Similar problems do not occur, for example, when computing deletion diagnostics—such as the Cook's distance—in classical linear (or generalized linear) models because of the existence of exact (approximate) update formulas for the required terms (Cook and Weisberg 1982). Approaches of this type have also been explored for a limited number of Bayesian problems (Carlin et al. 1992; Kass and Vaidyanathan 1992; McCulloch 1989; Tierney et al. 1989).

The methods we propose in this article have similar goals. We assume that a sample from the baseline posterior distribution of the model parameters can be generated through the Gibbs Sampler (Gelfand and Smith 1990). Expanding on ideas of Tanner (1991, p. 54), Gelfand et al. (1992) (who consider the issue of model determination from a predictive viewpoint), and Smith and Roberts (1993), we estimate the Kullback-Leibler divergence of this distribution from an alternative posterior distribution via a Monte Carlo approximation. The various terms in the approximation are functions of the likelihood ratios of the two distributions evaluated at the different points in the sample.

This approach has the desirable property that the *same* sample from the baseline posterior distribution can be used to estimate the Kullback-Leibler divergence from several alternative posteriors. Generation of one sample via the Gibbs Sampler may be computationally expensive in practical situations. By circumventing the need to redo the analysis for each alternative being considered, the proposed approach dramatically reduces the time needed to identify potentially influential observations and to probe the modeling assumptions.

*The author would like to thank Thomas Santner, Prem Goel, Steven MacEachern, and Nandini Raghavan for helpful comments and suggestions.

2 The Gibbs Sampler and the Gibbs Stopper

The Gibbs Sampler is a successive substitution sampling scheme that allows one to generate samples from the joint distribution of a set of random variables having density $g(\boldsymbol{x}) = g(x_1, \ldots, x_d)$ with respect to a dominating measure $\lambda(\boldsymbol{x})$ (Gelfand and Smith 1990; Tanner 1991). In Bayesian statistical applications $g(\boldsymbol{x}) = g(\boldsymbol{x}|\boldsymbol{y})$ will usually be the posterior probability density for the model parameters $\boldsymbol{x}$ conditional on the observations $\boldsymbol{y}$, and the samples will be used to estimate functionals of $g(\boldsymbol{x})$. The algorithm generates a path $\{\boldsymbol{x}^{(j)}\}$ of a Markov chain whose invariant probability distribution coincides with $g(\boldsymbol{x})$. Under mild regularity conditions the iterative scheme is guaranteed to converge in the sense that, if j is large enough, $\boldsymbol{x}^{(j)}$ can be regarded as a realization from $g(\boldsymbol{x})$ (Tierney 1991; Schervish and Carlin 1992).

Ritter and Tanner (1992) introduce a diagnostic criterion, called the *Gibbs Stopper,* to assess convergence in practical applications. Denote by $g_j(\boldsymbol{x})$ the density of the distribution of the chain at the j-th stage of the iterative procedure. If convergence has been attained, so that $g_j(\boldsymbol{x})$ is "close" to $g(\boldsymbol{x})$, then the ratio $g(\boldsymbol{x})/g_j(\boldsymbol{x})$ should be close to one over the whole range of possible $\boldsymbol{x}$ values. In general, the target density $g(\boldsymbol{x})$ will only be known up to a renormalization constant C, i.e. $g(\boldsymbol{x}) = C\gamma(\boldsymbol{x})$, and $g_j(\boldsymbol{x})$ will have to be estimated. Ritter and Tanner (1992) propose an estimate $\hat{g}_j(\boldsymbol{x})$ given by a Monte Carlo sum based on the two final sets of draws $\boldsymbol{x}_m^{(j-1)}$ and $\boldsymbol{x}_m^{(j)}$, $m = 1, \ldots, M$, from M independent paths of the Gibbs Sampler carried out to depth j.

Upon convergence the ratios

$$w_{g,m} = w_{g,m}^{(j)} = \frac{\gamma\left(\boldsymbol{x}_m^{(j)}\right)}{\hat{g}_j\left(\boldsymbol{x}_m^{(j)}\right)}, \quad \text{for } m = 1, \ldots, M, \qquad (1)$$

should be concentrated around a constant value. The Gibbs Stopper amounts to monitoring the ratios in Equation (1) and halting the algorithm once visual inspection of their histograms and evaluation of some functional of their distribution (e.g. their standard deviation) indicate that they have stabilized around a constant.

We will refer to the ratios in Equation (1) as *GS-weights.* Note that, for a fixed number of cycles j, by associating a probability $\bar{w}_{g,m} = w_{g,m}/\sum_{k=1}^{M} w_{g,k}$ to each of the points $\boldsymbol{x}_m^{(j)}$, $m = 1, \ldots, M$, one can regard them as a sample from $g(\boldsymbol{x})$ instead of $g_j(\boldsymbol{x})$ (Geweke 1989). In the sequel, when referring to a sample $\boldsymbol{x}_m$ from $g(\boldsymbol{x})$ obtained through M independent replicates of j cycles of the Gibbs Sampler, we will more precisely mean a sample $\boldsymbol{x}_m^{(j)}$ from $g_j(\boldsymbol{x})$ reweighted according to $\bar{w}_{g,m}$.

3 Monte Carlo Estimation of the Kullback-Leibler Divergence and Bayesian Analysis

Let two distributions have densities f and g with respect to a common dominating measure λ. The Kullback-Leibler divergence of g from f is defined as

$$\mathcal{K}(f,g) = \int \log\left(\frac{f(\boldsymbol{x})}{g(\boldsymbol{x})}\right) f(\boldsymbol{x})\, d\lambda(\boldsymbol{x}).$$

The use of the Kullback-Leibler divergence to evaluate discrepancies between distributions in an attempt to assess case deletion influence and sensitivity to prior assumptions is well documented in the statistical literature (Johnson and Geisser 1983; McCulloch 1989; Gelfand et al. 1992). Observe that, not being symmetric in its arguments, The Kullback-Leibler divergence is not a distance.

Throughout the section we will denote by $p(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y})$ the posterior density for the model parameters $\boldsymbol{x} = (x_1, \ldots, x_d)$ conditional on the set of observations $\boldsymbol{y} = (y_1, \ldots, y_n)$. We first discuss a comprehensive screening method for identifying influential observations within a Bayesian framework. Let I be a subset of the integers 1 through n and let $p_{\setminus I}(\boldsymbol{x}) = p_{\setminus I}(\boldsymbol{x}|\boldsymbol{y}_{\setminus I})$ be the posterior density for the model parameters $\boldsymbol{x}$ conditional on the reduced set of observations $\boldsymbol{y}_{\setminus I} = \{y_i : i \notin I\}$, Denote by $q(\boldsymbol{x}) = q(\boldsymbol{x}, \boldsymbol{y})$ and $q_{\setminus I}(\boldsymbol{x}) = q(\boldsymbol{x}, \boldsymbol{y}_{\setminus I})$ the joint densities of $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}, \boldsymbol{y}_{\setminus I})$ respectively. Then $p(\boldsymbol{x}) = q(\boldsymbol{x})/C$ and $p_{\setminus I}(\boldsymbol{x}) = q_{\setminus I}(\boldsymbol{x})/C_{\setminus I}$, with $C = \int q(\boldsymbol{x})\, d\lambda(\boldsymbol{x})$ and $C_{\setminus I} = \int q_{\setminus I}(\boldsymbol{x})\, d\lambda(\boldsymbol{x})$.

Suppose that a sample from $p(\boldsymbol{x})$ obtained through the Gibbs Sampler is available and that we wish to determine the effect that the presence or absence of individual observations has on our inferential conclusions by means of comparison between the posterior distribution p, conditional on the entire set of observations $\boldsymbol{y}$, and the n posterior distributions $p_{\setminus I}$, $I = \{i\}$, conditional on the n reduced subsets of observations obtained by deleting observation y_i in turn.

We propose to employ the available sample from $p(\boldsymbol{x})$ to compute Monte Carlo estimates of the n values of the Kullback-Leibler divergence $\mathcal{K}(p_{\setminus I}, p)$ of p from each of the $p_{\setminus I}$. This is done in an attempt to obtain a measure of the effect that inclusion of the i-th observation would have on our inferences. Large divergence values would suggest that the observation has small likelihood under the assumed model and was possibly generated by a stochastic mechanism that differs from the one generating the remainder of the dataset.

Suppose then that, having run M independent Gibbs paths to depth j for the full model, we have draws

$\boldsymbol{x}_m$ from $p(\boldsymbol{x})$ together with their associated GS-weights $w_{p,m}$, for $m = 1, \ldots M$. We show in Peruggia (1994) that a Monte Carlo estimate of the Kullbak-Leibler divergence $\mathcal{K}(p_{\setminus I}, p)$ is given by:

$$\mathcal{K}_{\setminus I} = \sum_{m=1}^{M} (\log w_{\setminus I,m}) \bar{w}_{p_{\setminus I},m} - \log \left(\sum_{m=1}^{M} w_{\setminus I,m} \bar{w}_{p,m} \right), \tag{2}$$

where

$$w_{\setminus I,m} = \frac{q_{\setminus I}(\boldsymbol{x}_m)}{q(\boldsymbol{x}_m)}, \qquad m = 1, \ldots, M. \tag{3}$$

While the numerical values of the Kullback-Leibler divergence can be used to express a quantitative judgment, the ratios in Equation (3) can be used to make a graphical assessment of influence. In fact, if $p(\boldsymbol{x})$ is close to $p_{\setminus I}(\boldsymbol{x})$, then the distribution of the M ratios should be concentrated around a constant value, which implies that the renormalized ratios

$$\bar{w}_m = \frac{w_m}{\sum_{k=1}^{M} w_k}, \qquad m = 1, \ldots, M, \tag{4}$$

should be concentrated around $1/M$. Examination of the box-plot of the set of weights in Equation (4), preferably after having applied a logarithmic transformation, can therefore be used to make a judgment.

These ideas generalize immediately to the case in which one is concerned with the influence that some aspects of the modeling assumptions (in particular prior specification) have on the inferential process (robustness and sensitivity analysis). Suppose a "baseline" specification of the model yields the joint density $q(\boldsymbol{x}, \boldsymbol{y}) = q(\boldsymbol{x})$ for the parameters $\boldsymbol{x}$ and the data $\boldsymbol{y}$. As before, we can run the Gibbs Sampler for this model and obtain M independent draws $\boldsymbol{x}_m$ from the posterior distribution $p(\boldsymbol{x})$ with their associated GS-weights $w_{p,m}$. Assume further that the modification of some aspects of the model leads to the alternative joint density $q_A(\boldsymbol{x}, \boldsymbol{y}) = q_A(\boldsymbol{x})$, with corresponding posterior density $p_A(\boldsymbol{x}|\boldsymbol{y}) = p_A(\boldsymbol{x})$ for the same parameter vector $\boldsymbol{x}$.

Then, once the set of ratios

$$w_{A,m} = \frac{q_A(\boldsymbol{x}_m)}{q(\boldsymbol{x}_m)}, \qquad m = 1, \ldots, M, \tag{5}$$

has been constructed, the analysis can proceed as before. In particular, we can examine the box-plot of the logarithms of the renormalized ratios to determine how concentrated they are, and we can estimate $\mathcal{K}(p_A, p)$ by:

$$\mathcal{K}_A = \sum_{m=1}^{M} (\log w_{A,m}) \bar{w}_{p_A,m} - \log \left(\sum_{m=1}^{M} w_{A,m} \bar{w}_{p,m} \right). \tag{6}$$

4 The Normal Means Model

We illustrate these ideas with an example. Consider the hierarchical Normal Means Model (Gelfand and Smith 1990). We observe L_k data points from the k-th of K normal populations, i.e. $y_{k,l} \sim N(\theta_k, \sigma_k^2)$, for $k = 1, \ldots, K$, and $l = 1, \ldots, L_k$. Conditional on the parameter values θ_k and σ_k^2, the observations are assumed to be independent within and between groups. Further, we assume the group means and variances to be independent with $\theta_k \sim N(\mu, \tau^2)$ and $\sigma_k^2 \sim IG(a_1, b_1)$ (a_1 and b_1 known). Finally, we assume μ and τ^2 to be independent with $\mu \sim N(\mu_0, \sigma_0^2)$, and $\tau^2 \sim IG(a_2, b_2)$ (μ_0, σ_0, a_2 and b_2 known). In the notation of the previous section, $\boldsymbol{x} = (\{\theta_k\}, \{\sigma_k^2\}, \mu, \tau^2)$, a $(2 \times K + 2)$-dimensional parameter vector, and $\boldsymbol{y} = \{y_{k,l}\}$.

We ran our experiment using *simulated* observations. Data $\boldsymbol{y}$ was generated from two independent normal populations ($K = 2$): the first sample, of size $L_1 = 10$, from a $N(0, 1)$ distribution, and the second, of size $L_2 = 8$, from a $N(0.5, 1)$ distribution. We completed the specification of the prior distributions by setting $\mu_0 = 0$, $\sigma_0 = 1$, $a_1 = a_2 = 4$, and $b_1 = b_2 = 0.333$. These choices imply that both σ^2 and τ^2 have mean 1 and variance 0.5. Based on these assumptions we performed the following influence and sensitivity analyses.

4.1 Influence

In order to illustrate how our method can be applied to detect influential observations we artificially introduced a spurious data point. Specifically, we shifted $y_{2,1}$ by 6 standard deviations to the left of its observed value of -0.067, setting it equal to -6.067. We then ran $M = 100$ independent Gibbs Sampler paths to depth $j = 200$, thus obtaining 100 draws $\boldsymbol{x}_m$ and associated GS-weights $w_{p,m}$ from the posterior distribution $p(\boldsymbol{x})$ conditional on the 18 observations, and assessed convergence using the Gibbs Stopper criterion of Section 2.

We then implemented the *leave-one-out* strategy for influence detection outlined in Section 3. In this setting, if we take $I = \{(k, l)\}$ (i.e. if we consider removing the l-th observation in the k-th group from the dataset), we obtain the following functional form for the ratios in Equation (3):

$$w_{\setminus I,m} = \frac{q_{\setminus I}(\boldsymbol{x}_m)}{q(\boldsymbol{x}_m)} = \frac{1}{\varphi(y_{k,l}|\theta_{k,m}, \sigma_{k,m}^2)}, \quad m = 1, \ldots, M, \tag{7}$$

where $\varphi(\cdot|\theta, \sigma^2)$ denotes the density function of a normal random variable with mean θ and variance σ^2. Adjacent box-plots of the 18 sets (corresponding to all $I = \{(k, l)\}$) of 100 ratios defined in Equation (7) (after renormaliza-

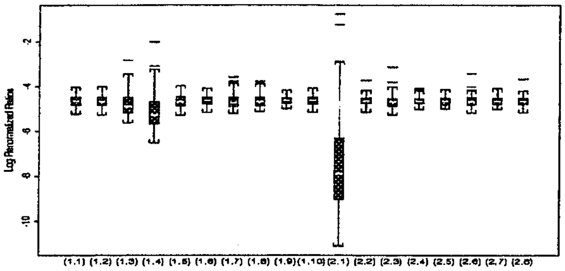


Figure 1: Box-Plots of the Logarithm of the Leave-One-Out Renormalized Ratios

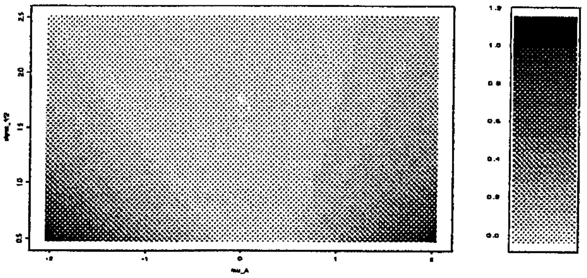


Figure 2: Sensitivity of Posterior to Prior Specification of $\left(\mu_0 = \mu_A, \sigma_0^2 = \sigma_A^2\right)$

tion and transformation on the logarithmic scale) are displayed in Figure 1.

As expected, the box-plot of the set of ratios corresponding to $I = \{(2,1)\}$ appears strikingly different from the others. In particular, in this case, there is a renormalized ratio as large as 0.465, while the overall maximum ratio over the remaining 17 sets belongs to the set corresponding to $I = \{(1,4)\}$ and equals 0.132. This suggests that suppressing observation $y_{(2,1)}$ from the data will exert a strong influence. More precisely, the posterior distributions for the parameters $\boldsymbol{x}$ given all 18 observations and given all 18 observations but the $(2,1)$-th will differ significantly. Visual inspection of the box-plots indicates that observations $y_{(1,4)}$ and $y_{(1,3)}$ may also be considered mildly influential.

Next we used Equation (2) to compute $\mathcal{K}_{\backslash I}$, the estimated Kullback-Leibler divergence of p from $p_{\backslash I}$, for all $I = \{(k,l)\}$. While the great majority of the estimated values are of the order of 10^{-2}, $\mathcal{K}_{\backslash\{(2,1)\}} \approx 2.8$, in strong agreement with the conclusions we had already drawn from visual inspection of the box-plots. Also in agreement with those conclusions is the fact that $\mathcal{K}_{\backslash\{(1,3)\}}$ and $\mathcal{K}_{\backslash\{(1,4)\}}$ are of the order of 10^{-1}. Peruggia (1994) contains a detailed analysis offering evidence of the considerable location shift and reduced variability in the marginal posterior density of θ_2 induced by the deletion of the outlying observation $y_{(2,1)} = -6.067$.

4.2 Sensitivity

Now we illustrate how the same approach can be employed to perform a sensitivity analysis. For the original dataset, we probed the effect of varying the prior specification of the mean μ_0 for the parameter μ as follows. Let $q(\boldsymbol{x})$ denote the joint density of $(\boldsymbol{x}, \boldsymbol{y})$ corresponding to the "baseline" specification of $\mu_0 = 0$. We considered 101 equally-spaced, alternative values $\mu_0 = \mu_A$ in the interval $[-5, 5]$. Each such value yielded a corresponding joint density $q_A(\boldsymbol{x})$ for $(\boldsymbol{x}, \boldsymbol{y})$. We used the Gibbs Sampler to generate $M = 100$ independent observations $\boldsymbol{x}_m$ and corresponding GS-weights $w_{p,m}$ from the posterior distribution having density $p(\boldsymbol{x}) = q(\boldsymbol{x})/C$, where $C = \int q(\boldsymbol{x})\, dx$. We then computed the 101 sets of ratios corresponding to each alternative μ_A according to Equation (5). More explicitly, with μ_m denoting the M values of μ generated via the Gibbs Sampler, we computed

$$w_{A,m} = \frac{q_A(\boldsymbol{x}_m)}{q(\boldsymbol{x}_m)} = \frac{\varphi(\mu_m|\mu_A, \sigma_0^2)}{\varphi(\mu_m|\mu_0, \sigma_0^2)}, \quad m = 1, \ldots, M,$$

and from these we derived, according to Equation (6), the estimated Kullback-Leibler divergence $\mathcal{K}_A$ of $p(\boldsymbol{x})$ from $p_A(\boldsymbol{x}) = q_A(\boldsymbol{x})/C_A$, where $C_A = \int q_A(\boldsymbol{x})\, dx$, for the 101 alternative values of μ_A being considered.

The plot of $\mathcal{K}_A$ versus μ_A was fairly symmetric around $\mu_0 = 0$, with a rate of increase only slightly higher for positive values of $\mu_0 = \mu_A$. Although the actual numerical values of the Kullback-Leibler divergence are difficult to interpret directly, it appeared that a prior specification of $\mu_0 = 0$ when the "true" value of μ_0 is some other value μ_A in the interval $[-2, 2]$ should not have an overwhelming impact on the resulting posterior distribution for the model parameters.

In a similar manner, and with little additional computational burden, it is possible to assess the effect of varying more than one prior parameter at a time. Figure 2 illustrates the results we obtained by altering simultaneously the values of μ_0 and σ_0^2 in the specification of the prior distribution of μ. At each point $\left(\mu_0 = \mu_A, \sigma_0^2 = \sigma_A^2\right)$, the figure displays the Kullback-Leibler divergence of the posterior distribution for $\boldsymbol{x}$ arising from the original specification $\left(\mu_0 = 0, \sigma_0^2 = 1\right)$ from the one arising from the alternative specifications $\left(\mu_0 = \mu_A, \sigma_0^2 = \sigma_A^2\right)$. Darker shades of gray correspond to larger divergence values, as indicated by the gray-scale bar on the right hand side of the figure. The display indicates clearly that a shift in the prior specification of μ_0 away from 0 has stronger repercussions on the inferential process for smaller values of σ_0^2.

It is intrinsically difficult to evaluate the numerical values of the Kullback-Leibler divergence on an absolute scale (see for instance McCulloch 1989). We looked at this problem from the point of view of equivalency between model specification and the presence of influential observations. Denote by p the posterior distribution for $\boldsymbol{x}$ conditional on all 18 observation $\boldsymbol{y}$ when $y_{(2,1)} = -6.067$, and by $p_{\setminus\{(2,1)\}}$ the posterior for $\boldsymbol{x}$ arising from the same model after removing $y_{(2,1)}$ from the analysis. We estimated before that $\mathcal{K}\left(p_{\setminus\{(2,1)\}}, p\right) = 2.817$. Observe that both p and $p_{\setminus\{(2,1)\}}$ are based on a prior specification of the parameter value $\mu_0 = 0$.

By employing the proposed Monte Carlo technique based on a random sample from $p_{\setminus\{(2,1)\}}$, we estimated that an alternative specification $\mu_A \approx -4.7$ of μ_0 would yield a posterior distribution $p_{\setminus\{(2,1)\},\mu_A}$ for which $\mathcal{K}\left(p_{\setminus\{(2,1)\}}, p_{\setminus\{(2,1)\},\mu_A}\right)$ is also approximately equal to 2.8. In other words, introducing the aberrant observation $y_{(2,1)} = -6.067$ into the analysis has the same effect on the posterior distribution for $\boldsymbol{x}$ (in term of Kullback-Leibler divergence) as moving the prior specification of μ_0 from 0 to -4.7. Thus, if we consider a shift from 0 to -4.7 in our prior beliefs about μ_0 to be important, we should also attach the same degree of relevance to the presence of the outlying observation $y_{(2,1)}$ in our dataset.

It is important in practical applications to be able to assess the Monte Carlo variance of the proposed estimates of the Kullback-Leibler divergence between two distributions. In Peruggia (1994) we discuss this issue and illustrate it within the context of the normal means model example.

Bibliography

Berger. J.O. (1985) *Statistical Decision Theory and Bayesian Analysis* (2nd ed.), Springer Verlag, New York.

Billingsley, P. (1986) *Probability and Measure* (2nd edition), John Wiley & Sons, New York.

Carlin, B.P., Kass, R.E., Lerch, F.J., and Huguenard, B.R. (1992) Predicting Working Memory Failure: A Subjective Bayesian Approach to Model Selection, *J. Am. Statis. Assoc.*, **87**, 319-327.

Cook, R.D., and Weisberg, S. (1982) *Residuals and Influence in Regression*, Chapman and Hall, New York.

DeGroot, M.H. (1986) *Probability and Statistics* (2nd edition), Addison-Wesley, Reading, Ma.

Gelfand, A.E., Dey, D.K., and Chang, H. (1992) Model Determination using Predictive Distributions with Implementation via Sampling-Based Methods, in *Bayesian Statistics 4*, 147-167.

Gelfand, A.E., and Smith, A.F.M. (1990) Sampling-Based Approaches to Calculating Marginal Densities, *J. Am. Statis. Assoc.*, **85**, 398-409.

Johnson, W., and Geisser, S. (1983) A Predictive View of the Detection and Characterization of Influential Observations in Regression Analysis, *J. Am. Statist. Assoc.*, **78** 137-144.

Geweke, J. (1989) Bayesian Inference in Econometric Models Using Monte Carlo Integration. *Econometrica,* **57**, No. 6 1317-1339.

Kass, R.E., and Vaidyanathan, S.K. (1992) Approximate Bayes Factors and Orthogonal Parameters, with Applications to Testing Equality of Two Binomial Proportions, *J. R. Statist. Soc. B*, **54**, 129-144

Kullback, S. (1959) *Information Theory and Statistics*, John Wiley, New York.

McCulloch, R.E. (1989) Local Model Influence, *J. Am. Statist. Assoc.*, **84**, 473-478.

Peruggia, M. (1994) Monte Carlo Assessment of Influence and Sensitivity in Bayesian Modeling via the Gibbs Sampler, *Department of Statistics - The Ohio State University*, Technical Report No. 544.

Ritter, C., and Tanner, M.A. (1992) Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy-Gibbs Sampler, *J. Am. Statis. Assoc.*, **87**, 861-868.

Schervish, M.J., and Carlin, B.P. (1992) On the Convergence of Successive Substitution Sampling, *J. Comp. Graph. Statist.*, **1**, 111-127.

Smith, A.F.M., and Roberts, G.O. (1993) Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods, *J. R. Statist. Soc.* **B**, **55**, No. 1, 3-23.

Smith, A.F.M., Skene, A.M., Shaw, J.E.H., and Naylor, J.C. (1987) Progress with Numerical and Graphical Methods for Practical Bayesian Statistics, *The Statistician*, **36** 75-82.

Tanner, M. A. (1991) *Tools for Statistical Inference*, Springer-Verlag, New York.

Tierney, L. (1991) Markov Chains for Exploring Posteriors Distributions, *School of Statistics - University of Minnesota*, Technical Report No. 560.

Tierney, L., and Kadane, J.B. (1986) Accurate Approximations for Posterior Moments and Marginal Densities, *J. Am. Statis. Assoc.*, **81**, 82-86.

Tierney, L., Kass, R.E., and Kadane, J.B. (1989) Approximate Methods for Assessing Influence and Sensitivity in Bayesian Analysis, *Biometrika*, **76**, 663-674.

Walker, A.M. (1969) On the Asymptotic Behaviour of Posterior Distributions, *J. R. Statist. Soc. B*, **31**, 80-88.

Using the Gibbs Sampler to Detect Changepoints: Application to PSA as a Longitudinal Marker for Prostate Cancer

Kathleen A. Cronin, Elizabeth H. Slate, Bruce W. Turnbull, Martin T. Wells
School of Operations Research/Industrial Engineering and Statistics Center
Cornell University, Ithaca, NY 14853

Abstract

We generalize the linear mixed-effects model introduced by Laird and Ware (1982) to include random changepoints, in a manner similar to Stephens (1994). We use a fully Bayesian hierarchical model in which the parametric forms are known between the changepoints and we estimate the changepoints and model parameters using Gibbs sampling. These techniques are applied to investigate prostate specific antigen (PSA) as a diagnostic indicator for prostate cancer by modeling longitudinal PSA measurements for which the changepoint is the onset of cancer. We are most concerned with the goal of accurate early detection. Diagnostic rules previously proposed in the medical literature are compared with measures based on the posterior probability of disease onset.

1 Introduction

Laird and Ware (1982) introduced a family of mixed-effects models which capture the serial correlation found in longitudinal data. We are interested in modeling longitudinal data where the underlying process changes at a random point in time. We extend the mixed-effects model to include this continuous random changepoint and use the Gibbs sampler to estimate model parameters including the changepoint.

There is a great deal of literature on identifying when a process has changed and estimating the changepoint. Page (1955) used non-parametric methods to test the hypothesis that all observations are from the same distribution. Hinkley (1969, 1970) used maximum likelihood estimation to identify a shift in process mean and the intersection of a two-phase regression. Smith (1975) presented a Bayesian approach to estimating changepoints for normal and binomial distributions along with an informal sequential procedure. Carlin et al. (1992) gave a fully Bayesian hierarchical analysis of changepoint problems, including the use of the Gibbs sampler to solve for the posterior distributions of model parameters. Stephens (1994) looked at continuously distributed changepoints and multiple changepoint identification from a retrospective point of view.

We describe a mixed-effects model with linear growth before and after the changepoint. We then apply the model to a simulated data set based on the longitudinal PSA measurements found in the study by Carter et al. (1992). We perform a prospective sequential analysis to see how quickly this method identifies a changepoint after it occurs and compare the results with other proposed diagnostic rules using receiver operator characteristic (ROC) curves.

2 Hierarchical model

The mixed-effects model for linear growth before and after the changepoint, t_i, can be written as

$$y_{ij} = a_{oi} + a_i x_{ij} + b_i \, (x_{ij} - t_i)^+ + \epsilon_{ij} \tag{1}$$

where y_{ij} is the measured value for subject i at observation j, and x_{ij} is the time of observation j for subject i. The index i takes values $1, \ldots, N$ and j takes values $1, \ldots, n_i$ when there are N subjects in the study and the ith has n_i observations. The complete model assumes the following distributions.

$$\begin{aligned}
\begin{pmatrix} a_{oi} \\ a_i \end{pmatrix} \Bigg| \begin{pmatrix} \alpha_o \\ \alpha \end{pmatrix}, \Sigma_a &\sim \mathrm{MVN}\left\{ \begin{pmatrix} \alpha_o \\ \alpha \end{pmatrix}, \Sigma_a \right\} \\
\begin{pmatrix} \alpha_o \\ \alpha \end{pmatrix} &\sim \mathrm{MVN}\left\{ \begin{pmatrix} \mu_{\alpha_o} \\ \mu_\alpha \end{pmatrix}, \begin{pmatrix} \sigma^2_{\alpha_o} & \sigma_{\alpha_o \alpha} \\ \sigma_{\alpha_o \alpha} & \sigma^2_\alpha \end{pmatrix} \right\} \\
\Sigma_a^{-1} &\sim \mathrm{Wishart}((\rho V)^{-1}, \rho) \\
b_i \mid \beta, \sigma_b^2 &\sim \mathrm{N}(\beta, \sigma_b^2) \\
\beta &\sim \mathrm{N}(\mu_\beta, \sigma_\beta^2)
\end{aligned} \tag{2}$$

$$\begin{aligned}
\frac{1}{\sigma_b^2} &\sim \mathrm{Gamma}(\lambda_b, r_b) \\
t_i | \tau, \sigma_t^2 &\sim \mathrm{N}(\tau, \sigma_t^2) \\
\tau &\sim \mathrm{N}(\mu_\tau, \sigma_\tau^2) \\
\frac{1}{\sigma_t^2} &\sim \mathrm{Gamma}(\lambda_t, r_t) \\
\epsilon_{ij} | \sigma_{\epsilon_i}^2 &\sim \mathrm{N}(0, \sigma_{\epsilon_i}^2) \\
\frac{1}{\sigma_{\epsilon_i}^2} &\sim \mathrm{Gamma}(\lambda_\epsilon, r_\epsilon)
\end{aligned}$$

The prior distributions for $\binom{\alpha_o}{\alpha}$, Σ_a, β, σ_b^{-2}, τ, σ_t^{-2}, $\sigma_{\epsilon_i}^{-2}$ are assumed known.

The Gibbs sampler, as described in Gelfand and Smith (1990), is used to solve for the posterior distributions of the model parameters. The procedure is similar to that in Lange et al. (1992), but with a continuous changepoint as described in Stephens (1994). The complete conditional distributions for each parameter, with the exception of the $\{t_i\}$, are standard parametric distributions and can be easily sampled. Although the form of the complete conditional distribution for t_i changes at each observation point, the form of the distribution between observation points is known. Hence the $\{t_i\}$ can be generated in a two step procedure which first generates an interval and then generates a point within that interval. Thus it is straightforward to generate from all the complete conditional distributions. This procedure leads to estimates of the subject specific parameters, including the $\{t_i\}$, based on posterior distributions. The hierarchical approach permits the "borrowing of strength" from the population to estimate the individual parameters while accounting for the within-subject serial correlation.

3 Application: Prostate disease and PSA

Prostate cancer is the second leading cause of cancer-related deaths among American males (Pearson et al. 1994). Garnick (1994) discusses the prevalence of prostate cancer and the dilemmas associated with diagnosis and treatment. There has been much controversy over the benefits and the possible dangers of screening for prostate cancer. In this application we do not address the larger question of whether screening should be performed, but look at a methodology that could be used to evaluate diagnostic rules used in screening.

Prostate specific antigen (PSA) is a glycoprotein produced by the prostate gland. The level of PSA found by a blood test increases with the volume of the prostate. The work of Catalona et al. (1991, 1993) supported the usefulness of PSA levels as a diagnostic marker for prostate cancer. Gerber (1991) discussed the value of screening along with a review of current screening methods. Oesterling et al. (1993) performed a prospective study to understand the link between PSA and age. He concluded that PSA increases gradually with age in normal men and suggested normal ranges of PSA for different age groups.

Carter et al. (1992), and Pearson et al. (1991, 1994) looked at serial PSA readings on men over a period of 7 to 25 years. They used a mixed-effects regression model to test whether the changes in PSA readings were different in men with and without prostate disease. Model parameters were estimated using a Newton-Raphson restricted maximum likelihood method. Carter et al. (1992) observed that PSA increases only very slowly with age before the onset of cancer and then increases more rapidly when cancer is present. As an approximating model, we will assume that it is the square root of the PSA level that follows the linear changepoint model (1).

Our work is motivated by longitudinal readings from the Nutritional Prevention of Cancer Trial (Abu-Libdeh et al. 1990, Clark et al. 1991). Over the course of the trial, participants have been giving blood at approximate six month intervals. Of these participants, some have developed prostate cancer. The principal investigator, Dr. L. C. Clark, plans to determine the PSA levels of the frozen blood samples from subjects with and without prostate cancer to further study the relationship between PSA levels and prostate disease.

We present results for an analysis based on simulated data. These data represent square root PSA measurements taken annually on 60 men over a 30 year period, with initial ages ranging from 28.4 to 89.6 years. First, random intercepts $\{a_{oi}\}$ and initial slopes $\{a_i\}$ were generated. Then, for 30 of these subjects ("cases") we simulated age-at-onset times by generating changepoints $\{t_i\}$ from a normal distribution with a mean of 70 years and a standard deviation of 10 years. For those subjects with changepoints, post-change slopes $\{b_i\}$ were generated. Finally, subject-specific and measurement errors were included to yield simulated square root readings $\{y_{ij}\}$. The parameters used for the simulation were derived from the longitudinal data presented by Carter et al. (1992).

We now analyze this simulated data set using the model described in Section 2. The prior distributions for $\binom{\alpha_o}{\alpha}$, Σ_a, β, σ_b^{-2}, τ, σ_t^{-2}, and $\sigma_{\epsilon_i}^{-2}$ are listed in the Appendix and are also based on the longitudinal study described in Carter et al. (1992). We take y_{ij} to be the square root of the PSA reading for subject i at observation j, and x_{ij} is the age of subject i at observation j.

We are primarily interested in sequentially estimating the marginal posterior distributions for the $\{t_i\}$ and the $\{b_i\}$. Figure 1 shows the trajectories for the square root PSA readings for one of the 30 simulated cases as it evolves over time. This subject's initial reading was at age 48.3 years and the changepoint occurred at age 65.5. Figure 2 shows the evolution of the posterior distribution of the changepoint t_i for this subject.

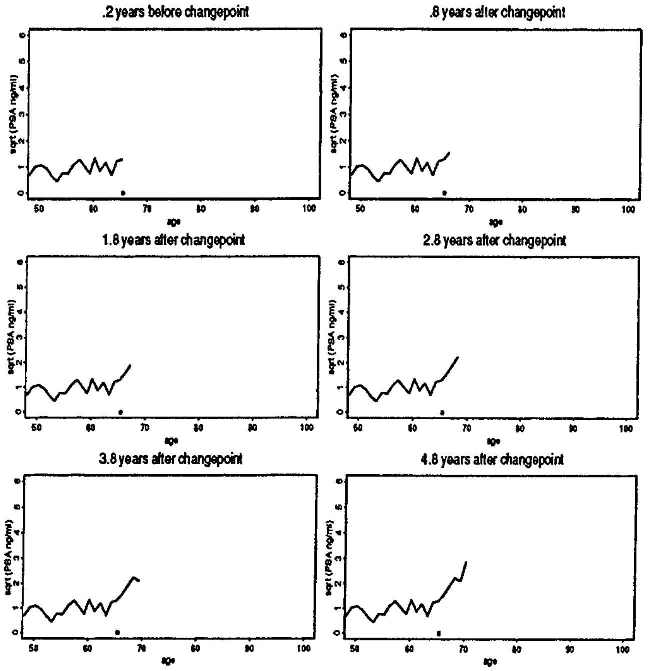


Figure 1: Trajectory of a typical simulated case. Dot at age 65.5 years indicates the changepoint

4 Comparison of diagnostic rules

Three different diagnostic rules or criteria have been suggested for use in screening for prostate cancer (Carter et al. 1992). The first is based on a normal range, whereby any PSA reading above a threshold value (typically 4 ng/ml) is considered a positive test result. The second and third diagnostic rules are based on a rate of increase over a given time period (e.g. 1.0 ng/ml/year over one year and .75 ng/ml/year over a two year period). The formulation we have proposed leads naturally to a fourth rule. At the time of the current test for a particular subject, we compute the posterior probability that the changepoint has already occurred. If the probability exceeds some specified cutoff value, then a positive result is indicated. We would like to compare these four suggested criteria — threshold, one year increase, average two year increase and posterior probability.

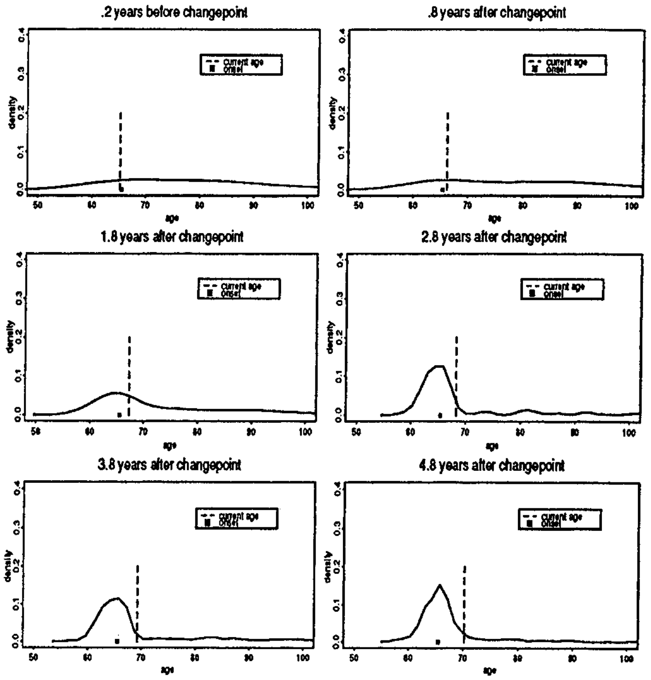


Figure 2: Posterior distributions of the changepoint t_i for the case illustrated in Figure 1.

A standard method of comparing diagnostic rules is to use receiver operator characteristic (ROC) curves (Centor, 1991). ROC curves plot sensitivity versus (1-specificity) as the cutoff value for the given criterion varies. Specificity is defined as the proportion of non-diseased subjects that test negative, and sensitivity as the proportion of diseased subjects that test positive. These definitions were developed for a single test and do not directly apply to a sequence of tests taken periodically over time. This is because, with longitudinal data, a single subject can be classified as a false positive at one observation time and as a true positive at a later observation time. Murtaugh et al. (1991) discussed ROC curves for repeated markers. They classified each subject as either true positive, false positive, true negative or false negative using the series of observations, thus effectively reducing the problem to the single test case.

We define a specificity rate, *spec*$_i$, for subject i as

$$\text{spec}_i = \frac{\text{number of negative tests before changepoint}}{\text{number of tests before changepoint}}.$$

An estimate of population specificity is obtained by averaging the subjects' rates. This definition weights each subject in the sample equally and incorporates all the data available.

We use a different approach to define sensitivity than we do specificity for two reasons. The first is that sensitivity is time dependent, a negative result ten years after the changepoint cannot be compared with a negative result within two years of the changepoint. Second, a true positive result ends the series of observations. This leads us to define a sensitivity indexed by time, *K-period sensitivity*, where a period is the time between tests. Here, for convenience, we assume the same period for all subjects. A true positive is a subject with any positive test result within K periods after the changepoint, and a false negative is a subject with no positive test results within K periods after the changepoint. K-period sensitivity is the proportion of diseased subjects that test positive *at any time* within K periods after onset.

We now use these definitions to compare the four diagnostic rules. We construct ROC curves using our simulated data for which the period is one year. Figure 3 shows ROC curves for four different values of K (K=1,2,3,4). The curves show that the threshold criterion is inferior, but that the others perform similarly two or more years after the changepoint. In practice, one may choose a rule by first identifying an acceptable level for specificity and then selecting the rule with the highest sensitivity. For our simulated data, the posterior probability achieves the highest sensitivity for specificity values greater than 95 percent.

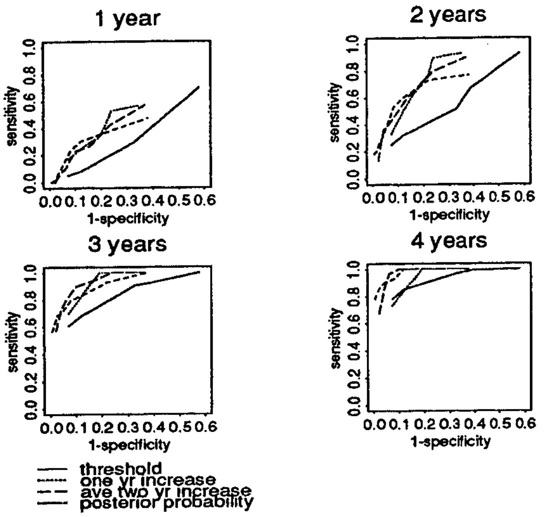


Figure 3: ROC Curves For Simulated Data

Acknowledgments

The authors would like to thank Dr. Larry Clark, University of Arizona Cancer Center, for suggesting the application to PSA and prostate cancer. The first author was supported by a National Defense Science and Engineering Graduate Fellowship. The research of Slate, Turnbull and Wells was supported by Grant PHS R01 CA61120; the research of Turnbull was also supported by Grant PHS R01 GM28364. Both grants are from the U.S. National Institutes of Health.

References

Abu-Libdeh, H., Turnbull, B.W. and Clark, L.C. (1990). "Analysis of multi-type recurrent events in longitudinal studies: Application to a skin cancer prevention trial." *Biometrics*, **46**, 1017-1023.

Carlin, B.P., Gelfand, A.E. and Smith, A.F.M. (1992). "Hierarchical Bayesian analysis of changepoint problems." *Appl. Statist*, **41**, No. 2, 389-405.

Carter, H.B., Pearson, J.D., Metter, E.J., Brant, L.J., Chan, D.W., Andres, R., Fozard, J.L. and Walsh, P.C. (1992). "Longitudinal evaluation of prostate-specific antigen levels in men with and without prostate disease." *J. Amer. Med. Assoc.*, **267**, No. 16, 2215-2220.

Catalona, W.J., Smith, D.S., Ratliff, T.L., Dodds, K.M., Coplen, D.E., Yuan, J.J., Petros, J.A. and Andriole, G.L. (1991). "Measurement of prostate-specific antigen in serum as a screening test for prostate cancer." *New Engl. J. Med.*, **324**, No. 17, 1156-1161.

Catalona, W.J., Smith, D.S., Ratliff, T.L. and Basler, J.W. (1993). "Detection of organ-confined prostate cancer is increased through prostate-specific antigen-based screening." *J. Amer. Med. Assoc.*, **270**, No. 8, 948-954.

Centor, R.M. (1991). "Signal detection: the use of ROC curves and their analyses." *Medical Decision Making*, **11**, No. 2, 102-106.

Clark, L.C., Patterson, B.H., Weed, D.L. and Turnbull, B.W. (1991). "Design issues in cancer chemoprevention trials using micronutrients: application to skin cancer." *Cancer Bulletin*, **43** No. 6, 519-524.

Garnick, M.B. (1994). "The dilemmas of prostate cancer." *Scientific American*, **270**, No. 4, 72-81.

Gelfand, A.E. and Smith, A.F.M. (1990). "Sampling-based approaches to calculating marginal densities." *J. Amer. Statist. Assoc.*, **85**, No. 410, 398-409.

Gerber, G.S. and Chodak, G.W. (1991). "Routine screening for cancer of the prostate." *J. Nat. Cancer Inst.*, **83**, No. 5, 329-335.

Hinkley, D.V. (1969). "Inference about the intersection in two-phase regression." *Biometrika*, **56**, No. 3, 495-504.

Hinkley, D.V. (1970). "Inference about the change-point in a sequence of random variables." *Biometrika*, **57**, No. 1, 1-17.

Laird, N.M. and Ware, J.H. (1982). "Random-effects models for longitudinal data." *Biometrics*, **38**, 963-974.

Lange, N., Carlin, B.P. and Gelfand, A.E. (1992). "Hierarchical Bayes models for the progression of HIV infection using longitudinal CD4 T-cell numbers." *J. Amer. Statist. Assoc.*, **87**, No. 419, 615-632.

Murtaugh, P.A., Wieand, H.S. and Schaid, D. (1991). "Application to ROC curve methodology when markers are repeated measures." Paper presented at Spring Meeting of The Biometric Society (ENAR), Houston Texas, March 24-27.

Oesterling, J.E., Jacobsen, S.J., Chute, C.G., Guess, H.A., Girman, C.J., Panser, L.A. and Lieber, M.M. (1993). "Serum prostate-specific antigen in a community-based population of healthy men." *J. Amer. Med. Assoc.*, **270**, No. 5, 660-664.

Page, E.S. (1955). "A test for a change in a parameter occurring at an unknown point." *Biometrika*, **42**, 523-527.

Pearson, J.D., Kaminski, P., Metter, E.J., Fozard, J.L., Brant, L.J., Morrell, C.H. and Carter, H.B. (1991). "Modeling longitudinal rates of change in prostate specific antigen during aging." *1991 Proceedings of the Social Statistics Section of the American Statistical Association*, Washington, D.C.

Pearson, J.D., Morrell, C.H., Landis, P.K., Carter, H.B. and Brant, L.J. (1994). "Mixed-effects regression models for studying the natural history of prostate disease." *Statistics in Medicine*, **13**, 587-601.

Smith, A.F.M. (1975). "A Bayesian approach to inference about a change-point in a sequence of random variables." *Biometrika*, **62**, No 2, 407-416.

Stephens, D.A. (1994). "Bayesian retrospective multiple-changepoint identification." *Appl. Statist*, **43**, No. 1, 159-178.

Appendix: Prior Distributions

We list here the prior distributions used for the application described in Section 3.

$$\begin{aligned}
\begin{pmatrix} \alpha_o \\ \alpha \end{pmatrix} &\sim \text{MVN}\left\{\begin{pmatrix} 1 \\ .04 \end{pmatrix}, \begin{pmatrix} .1 & 0 \\ 0 & .0001 \end{pmatrix}\right\} \\
\Sigma_a^{-1} &\sim W\left\{\begin{pmatrix} .1 & 0 \\ 0 & .0001 \end{pmatrix}^{-1}, 2\right\} \\
&= W((\rho V)^{-1}, \rho) \\
\beta &\sim \text{N}(.4, .01) \\
\frac{1}{\sigma_b^2} &\sim \text{Gamma}(3, .03) \\
\tau &\sim \text{N}(70, 25) \\
\frac{1}{\sigma_t^2} &\sim \text{Gamma}(3, 675) \\
\frac{1}{\sigma_{\epsilon_i}^2} &\sim \text{Gamma}(3, .27)
\end{aligned}$$

Applied Convergence Diagnostics for the Gibbs Sampler

Angelo Canty
Department of Statistics
University of Toronto

Abstract

One of the most difficult aspects of using the Gibbs sampler in practice is knowing when to stop the algorithm. In order to answer this we need to have some method which will tell us when we have completed enough iterations for the chain to have converged sufficiently. In this paper I will look at some of the methods that have been suggested in the literature. Most of these methods require input from the user throughout the length of the chain. This aspect of the diagnostics extends the length of time that it takes for the algorithm to terminate and is quite tedious for the user. Ideally one would like to have an automatic algorithm which would test for convergence and stop the Gibbs sampler when it is sufficiently close to convergence. I will look at some of the issues involved in finding such a diagnostic.

1 Introduction

Markov Chain Monte Carlo (MCMC) methods have recently become very popular tools for the analysis of Bayesian posterior distributions of relatively high dimension. The simplest of these algorithms is the Gibbs Sampler which was introduced by Geman and Geman (1984) in the context of image processing. It was then applied to Bayesian problems by Gelfand and Smith (1990) and Gelfand et al (1990). With this method we set up a Markov chain which has the posterior distribution of interest as its stationary distribution. Then by running the chain long enough we can sample from the posterior and so make inferences about it by simulation.

The major problem with the application of the Gibbs sampler is that it is very hard to know when the chain is sufficiently close to the target distribution for us to use it for inference. In some cases it is possible to calculate bounds on the total variation distance between the distribution of the chain after n iterates, $\pi^{(n)}$, and the target distribution, π. Then we can find out *a priori* how many iterations we need in order to make this distance as small as we like. At present, however, such methods have proved successful only in a very limited class of mathematically tractable models. Also the bounds are often quite loose and so can seriously over-estimate the number of iterations required to convergence. For examples of this method see Rosenthal (1991, 1993, 1994) and Meyn and Tweedie (1993).

A more applied approach to the problem is to use the output of the Gibbs sampler itself to assess when the chain is close to its target distribution. It is only necessary to run one implementation of the Gibbs sampler for the theoretical convergence results of MCMC to hold, however, it is often very difficult to differentiate convergence from transient behaviour based on a single run. Gelman and Rubin (1992) gave an example of the Ising model in which they ran two chains from different starting values. Individually, each chain appeared to have converged well after 2000 iterations but the two chains appeared to have converged to different distributions. Since the stationary distribution is unique for the Gibbs sampler, it is clear that the chains had not actually converged. For this reason I believe that it is essential for applied Gibbs sampling that a number of independent chains, each with the required stationary distribution, are used. Then, if only the final iterates from each chain are used, we have an *iid* sample from the target distribution. Even with multiple chains, the problem remains that one is trying to assess convergence of a sequence of d-dimensional distributions based on a finite sample.

2 Existing Methods for Assessing Convergence

The first method that was proposed to assess convergence was the *Thick Pen* method (Gelfand et al. 1990). In this method the user plots successive density estimates of the univariate variable of interest and claims that convergence has been achieved when the density estimates differ by only a very small amount. Although this method appeared to work for some models it was clear that a better diagnostic was needed for the more complex problems which it was hoped the Gibbs sampler would be applied to.

More recent diagnostics include the *Gibbs Stopper* (Ritter and Tanner 1992). This method was one of the first methods to try to assess convergence in $\Re^d$. The method is based on importance sampling. After n iterates of the m chains we estimate the current approximation $\hat{\pi}^{(m)}$ to the target distribution π. We then find the importance weights evaluated at the n^{th} iterates of each chain.

$$w_i^{(n)} = \frac{\hat{\pi}^{(n)}(X_i^{(n)})}{\pi(X_i^{(n)})}$$

The chains are assessed to have converged when the distribution of the weights is close to a spike at 1 (or some constant if π is not normalized). This method has a number of disadvantages. First, it requires knowledge of the normalizing constants in the full conditional densities or at the very least a good estimate of them. In most non-conjugate models such constants are not known and the estimation process is very time consuming. Secondly, the assessment of convergence is very subjective and requires that the user monitor the weight distribution for quite a while before being able to say that the weight distribution is close to a spike. Finally, the code for this method is highly dependent on the densities involved. Hence it is necessary to write new code each time a new model is used.

Another very popular convergence diagnostic was proposed by Gelman and Rubin (1992). This method looks at convergence of 1-dimensional variables of interest. In this method each of the m independent chains are allowed to run $2n$ iterations. The first n iterates are then discarded and we just look at the variable of interest in the second n iterates of each chain. An ANOVA type analysis is then applied to this $m \times n$ matrix of observations. The algorithm then calculates the within chain and between chain variances as well as estimates of the overall mean and variance. This method assumes that the variable of interest is approximately normally distributed so a conservative Students t distribution is used to give the current estimate of this distribution. Finally we find the potential scale reduction if sampling was allowed to continue to infinity. When this value is close to 1 convergence is said to have been achieved. This method has the advantage that we get a numerical value which is easier to assess. Also generic code is available which allows the method to be used on the output of any Gibbs sampler. One drawback is the assumption of approximate normality of the variable of interest and the cases that are of more applied interest are those where the assumption of normality is not justified.

The most mathematically sound convergence diagnostic was proposed by Roberts (1992). This method requires that we run a reversible Gibbs sampler which in one iteration cycles from the first component of X to the last and then from the last component back to the first again. Based on this sampler Roberts defines a distributional norm such that $||\pi^{(n)} - \pi|| \downarrow 0$. He then constructs an unbiased estimator of $1 + ||\pi^{(n)} - \pi||$. The sequence of true values being estimated is a monotone sequence with 1 as its limit, hence by looking at the estimates after each iteration we should be able to see if convergence is indicated. Note that if the normalizing constant for π is not known then the convergence is to an unknown constant. The major problem with this method is that the estimator can have very high variance which often masks the monotone convergence of the quantity it is estimating. We must also know the normalizing constants for the full conditionals or find good approximations to them in order to calculate the estimate. Once again this method requires that new code be written for each new problem.

All of these methods require some sort of subjective assessment by the user as to when the chain has reached convergence. In practice this means that the user must monitor these diagnostics while the Gibbs sampler is running. Due to the often slow convergence of the Gibbs sampler this can require a lot of interaction between the user and the algorithm and so take a lot of user time and also slow down the running time for the Gibbs sampler. In the next section I will look at whether it is possible to reduce this user interaction in the convergence assessment process.

3 Automating the Termination Procedure

Ideally one would like a totally automatic procedure which would run the Gibbs sampler without any user interaction until the chains had converged to the target distribution and then return a sample from this distribution. This requires a convergence diagnostic which can be monitored for signs of convergence by a computer algorithm without any user involvement. Clearly such a convergence diagnostic would need to be totally numerical and there would need to be some test of when the value of the convergence diagnostic indicates convergence. Unfortunately, such a totally automatic algorithm does not seem to be possible. It is very easy for the Gibbs sampler to become stuck in areas of the sample space which have a local mode for long periods of time and if all the chains should happen to become stuck at the same mode then any convergence diagnostic would indicate convergence even though it had not occurred yet. It should be possible, however, to reduce user interaction by having an algorithm which could detect when convergence has not occurred and only ask for user input when the diagnostic indicates that convergence may have been achieved.

Suppose that the user runs m independent chains from an initial distribution. The initial distribution should be over-dispersed relative to the target distribution so that important areas of the target distribution are not missed. Let Z be a 1-dimensional variable of interest which is a function of X. Then the goal is to have the algorithm continue sampling when the current distribution of the variable of interest is not the true target distribution of Z, and to alert the user when it may be at the correct distribution.

If we assume that we do not start the chains from the stationary distribution but from some other distribution, then convergence cannot have been achieved while the chains are still sampling from the initial distribution. Therefore the first part of the proposed method is to continue sampling until the chains appear to have left the initial distribution. In order to test this we need only look at the initial sample and the current sample of final values from each chain. We then need a way of comparing the distributions which produced these two samples. Since the distribution of $Z^{(n)}$ is unknown we need a non-parametric test for the equality of two distributions. The usual tests do not appear to have enough power to detect the small differences that are possible and so I have constructed another test. This test is more powerful than the standard tests as long the effective support of the two distributions are equal.

The test, for the two samples $Z_1^{(0)}, \ldots, Z_m^{(0)}$ and $Z_1^{(n)}, \ldots, Z_m^{(n)}$ is as follows.

1. Fit a line to the $q - q$ plot of the two samples. Let $\left(\hat{\alpha}, \hat{\beta}\right)$ be the estimates of the intercept and slope respectively.

2. Under the hypothesis that the two samples are from the same distribution, the true values of (α, β) are $(0, 1)$. Hence a measure of the difference between the distributions would be the distance between the estimates and $(0, 1)$. This distance is given by

$$\left|(\hat{\alpha}, \hat{\beta})\right|^2 = \hat{\alpha}^2 + (\max(\hat{\beta}, 1/\hat{\beta}) - 1)^2.$$

3. Now take b pairs of bootstrap samples from the sample $\left\{Z_1^{(n)}, \ldots, Z_m^{(n)}\right\}$ and repeat the first two steps for each pair. Since each pair is a pair of samples from the distribution of $Z^{(n)}$, this will estimate the variability of the distance measure if the two samples come from the same distribution.

4. If less than 5% of the bootstrap distances are larger than the observed distance then we can conclude that the chains have left the initial distribution. If more than 5% of the bootstrap distances are greater than our observed distance continue sampling.

It is inefficient for the algorithm to test departure from the initial distribution after every iteration so it is recommended that it do the test every *gap* iterates where *gap* is a user supplied integer. The ideal value for *gap* will depend on the initial distribution and the true distribution of Z.

Once the algorithm has found n such that the distribution of $Z^{(n)}$ is significantly different from the distribution of $Z^{(0)}$ it can start testing for convergence. In order for convergence to have been achieved, it is necessary that all m chains be sampling from the true distribution of Z. Also under convergence the across chain distribution should be the true distribution. The algorithm that I propose will test if all chains are sampling from the same distribution and if this is also the across chain distribution. It will not, however, test if this is the true distribution. Such a test would require that complex code be written for every new model and every new variable of interest. If all chains are sampling from the same distribution then it is probable that it is the true distribution but the assessment of this is left to the user's knowledge of the actual model.

In order to look for possible convergence I propose that the chains be allowed to run a further n iterations to give a total of $2n$ iterates. Then we can compare the m within chain distributions to the across chain distribution at time $2n$. We will use the same distance measure as before, so we must take a sample of size m from each chain. These m observations should be taken from the second half of the chain. I have used equally spaced iterates between $n + 1$ and $2n$. Compare each of these m samples with the sample $\left\{Z_1^{(2n)}, \ldots, Z_m^{(2n)}\right\}$. Then define the maximum squared distance to be the maximum of the m squared distances. For the bootstrap part take the pairs of bootstrap samples from the across chain sample at time $2n$. The bootstrap distribution for the squared distance measure for samples from $\pi^{(2n)}$ is all that is needed to test convergence since the m chains are independent and under the null hypothesis they are samples from the same distribution as $Z^{(2n)}$.

Hence, under the null hypothesis the squared distances are *iid*, and so the quantiles of their maximum can be derived from the quantiles of the distribution of squared distances for pairs of samples from $\pi^{(2n)}$. Therefore the null hypothesis of convergence should be rejected if the observed p-value is less than $1 - \sqrt[m]{0.95}$.

If the null hypothesis of convergence is rejected at iteration $2n$, then the algorithm should automatically continue with sampling. For small n it is possible that the null hypothesis was rejected because the chains did not have sufficient time to move over the whole sample space. Therefore, the algorithm should now double the length of the chain again and in the next test use the final $2n$ iterates. Each time the chain length doubles the algorithm needs to store twice as many values so for the purposes of efficiency and physical storage the distance between tests cannot be doubled indefinitely. I propose that the chain length be doubled after each test until the time between tests is large enough that under convergence the chains should move over the whole sample space in that number of iterates. Clearly this will depend on the model that is being used and also will be determined by the storage capacity of the machine. For this reason I feel that this maximum number of iterates between tests should be a user supplied value.

Once the hypothesis of convergence is not rejected, there is no more that the algorithm can say at that point. The user should then be notified that convergence *may* have been achieved. At this point it is up to the user to see whether convergence has actually been achieved or if the chains are simply stuck in a portion of the sample space. The easiest way of doing this is to plot the sample paths of each chain over the iterates on which the final convergence test was based. All of these plots should be similar and they should all cover the important areas of the sample space. At this point one could also try the Gelman and Rubin test on the same matrix of observations as used by the final test. If both of these user checks seem to confirm the result that convergence has been achieved then it is probably safe to use the chains for inference.

For this method, and most convergence diagnostics, to succeed it is very important that the initial distribution be chosen to cover the complete effective sample space. If all of the chains are started near a local mode then it is likely that the algorithm will assess convergence much sooner than is correct since each of the chains will tend to stay near the local mode and so all the chains will have the same distribution as the across chain sample but it not the correct target distribution. This situation can also arise due to chance if the starting values are selected at random. For this reason it is vital that the user have some idea of what the sample space is and to make sure that this whole area is covered by all of the chains when the algorithm does not reject the hypothesis of convergence. One way of avoiding a bad random sample is to select the starting points systematically to cover an area which is larger than the sample space of the target distribution. This is the method that many users of Gibbs sampling actually use in practice to find starting values for the chains. The proposed algorithm will still work for starting values chosen in this way, the only difference being that for such starting values deviation from the initial distribution would be detected sooner and so the second phase of the algorithm would start earlier than for starting values chosen from a distribution which approximates the target better. This difference does not appear to affect the number of iterations before the algorithm detects possible convergence.

Since the algorithm is totally free of any distributional assumptions, code can be written to do the testing which can then be applied to any situation. All that is required is the ability to incorporate the code for the convergence testing into the code for the Gibbs sampling, and there will need to be some global variables which keep track of whether the algorithm is testing for deviation from the initial distribution or testing for convergence, and when the next test is due. The extra time that is used by the algorithm to complete the required tests is not prohibitive to its use in practice as long as *gap* is chosen well and the number of bootstrap samples is not excessive. In most cases I have found that about 1000 bootstrap pairs is sufficient.

4 Examples

Here I will present 2 examples on which I used this algorithm. Both of them are examples where previous convergence diagnostics have had trouble. In both of these cases I ran 25 independent chains and for each test I used 1000 pairs of bootstrap samples. For the second stage of the convergence test the algorithm requires an observed p-value of greater than $1 - \sqrt[25]{0.95} = 0.0021$ in order to not reject the hypothesis of convergence.

Example 1

For the first example I used an equal mixture of bivariate normals. The distributions were centered at $\mu_1 = (0,0)$ and $\mu_2 = (4,4)$. In both cases the covariance matrix was the identity matrix. The variable of interest was the first component of X. For starting values I took 25 equally spaced points along the line $x = y$ between (-4,-4) and (8,8). For the first stage of the algorithm I tested deviation from the initial distribution every 50

Table 1: Convergence test for example 1

Iteration	Max Squared Distance	Bootstrap p-value
200	8.447	0
400	5.916	0.002
800	6.307	0
1600	2.872	0.009

iterations.

It took only 100 iterations for the algorithm to detect deviation from the initial distribution and then took a further 1500 iterations until the first time that convergence was indicated. Plots of the final 800 sampled values of the variable of interest showed that all chains moved between the two modes with approximately the correct frequencies at each mode. The convergence tests are summarized in table 1. In this example the algorithm quickly detected deviation from the initial distribution, but this simply meant that the points were no longer evenly spread. All that had happened by 100 iterations was that the chains had moved towards the closest mode and so there were two groups of chains. It then took a relatively long time for all the chains to move to convergence.

Example 2

For the second example I used the "Witch's Hat" distribution given by

$$\pi(x) = \frac{1-\delta}{(\sqrt{2\pi}\sigma)^d} \exp\left\{-\frac{\sum(x_i-\mu_i)^2}{2\sigma^2}\right\} + \delta \mathrm{I}_{[0,1]^d}$$

For this example I took $d = 8, \mu_i = 0.7$; $i = 1, \ldots, d$, $\sigma = 0.03$ and $\delta = 10^{-11}$. This distribution has a very sharp peak at the point μ and a flat brim on the rest of the set $[0, 1]^d$. Since the effective sample space is $[0, 1]^d$, I used the uniform distribution over this set as my initial distribution. The variable of interest that I looked at was the first component of X again.

In this case it took 600 iterations before the algorithm detected deviation from the initial uniform distribution. The results of the second stage of the algorithm are in table 2. Convergence is suggested after only two tests in this case and sample path plots of the first component from iteration 1201 to 2400 showed that all 25 chains were sampling from the spike and so convergence could be assumed.

Table 2: Convergence test for example 2

Iteration	Max Squared Distance	Bootstrap p-value
1200	39.9537	0
2400	0.3588	0.052

References

Gelfand, A.E., Hills, S.E., Rancine-Poon, A., and Smith, A.F.M. (1990) "Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling," *J. Amer. Stat. Assoc.*, **85**, 972 — 985.

Gelfand, A.E., and Smith, A.F.M. (1990), "Sampling-Based Approaches to Calculating Marginal Densities," *J. Amer. Stat. Assoc.*, **85**, 398 — 409.

Gelman, A., and Rubin, D.B. (1992), "A Single Series from the Gibbs Sampler Provides a False Sense of Security," In *Bayesian Statistics 4* (J.M. Bernardo, J.O. Berger, A.P. David, and A.F.M. Smith, editors), 625 — 632, Oxford University Press.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **6**, 721 — 741.

Meyn, S.P., and Tweedie, R.L. (1993), "Computable Bounds for Convergence Rates of Markov Chains," Technical Report, Department of Statistics, Colorado State University.

Ritter, C., and Tanner, M.A. (1992), "Facilitating the Gibbs Sampler: The Gibbs Stopper and the Griddy Gibbs Sampler," *J. Amer. Stat. Assoc.*, **87**, 861 — 868.

Roberts, G.O. (1992), "Convergence Diagnostics of the Gibbs Sampler" In *Bayesian Statistics 4* (J.M. Bernardo, J.O. Berger, A.P. David, and A.F.M. Smith, editors), 775 — 782, Oxford University Press.

Rosenthal, J.S. (1991), "Rates of Convergence for Gibbs Sampling for Variance Components Models," Technical Report 9322, Department of Statistics, University of Toronto.

Rosenthal, J.S. (1993), "Minorization Conditions and Rates of Convergence for Markov Chain Monte Carlo," Technical Report 9321, Department of Statistics, University of Toronto.

Rosenthal, J.S. (1994), "Analysis of the Gibbs Sampler for a Model Related to James-Stein Estimators," Technical Report 9413, Department of Statistics, University of Toronto.

STATISTICAL INFERENCE FOR PRIORITY QUEUES

Sudha Jain*
Department of Statistics
University of Toronto, Toronto
Ontario, Canada M5S 1A1

R. K. Jain
Department of Mathematics And Statistics
Memorial University of Newfoundland
St. John's, Newfoundland, Canada A1C 5S7

ABSTRACT

In this paper, the likelihood method for confidence intervals estimation of traffic intensity for the M/M/1 queueing system is investigated for the priority queues. The approximate confidence interval formulae of the mean queue lengths are derived for the M/M/1 priority queueing system. Numerical examples illustrate the above techniques.

1. Introduction

The idea of statistical analysis to queueing data dates back to Clarke's (1957) paper where he estimated the parameters for a simple M/M/1 queueing system using the principles of the maximum likelihood. Later, Lilliefors (1966) examined the problem of finding the confidence interval for the traffic intensity ρ ($\rho = \frac{\lambda}{\mu}$ is the ratio of mean arrival rate to mean service rate). He used the estimates of traffic intensity to obtain the confidence intervals for the expected number of units in the system. A direct approach based on the number of arrivals during the nth service period for the $M/E_k/1$ queue discussed by Bhat and Rao (1987), was used by Jain (1991) to obtain confidence intervals for the traffic intensity ρ. This paper addresses the problem of priority queue in which the service discipline is first come, first served; however the service could be interrupted if a customer with priority arrived. Obviously, the problem of modelling priority queues are generally more difficult to handle [see Gross and Harris (1985)]. However, it is that the real-life queueing situations for priority considerations are often required (viz. in an emergency department of a hospital and post office etc.).

In this paper, we consider the situation where the highest priority customer is allowed to enter the service immediately even if another customer with lower priority is already present in the service when the higher priority customer arrives to the system. Such system is called preemptive priority queueing system [see Gross and Harris (1985)]. The object of this paper is to estimate the parameters of such a system and then to obtain confidence intervals formulae of expected number of queue length for the priority and nonpriority customers. In section 2, preliminary results concerning the estimation of parameters for M/M/1 queueing systems are given. Section 3 deals with the parameters estimation procedures for the priority and non-priority customers. The approximate confidence intervals formulae for the mean queue length are obtained for the M/M/1 priority queues. Finally, numerical procedures are illustrated with an example.

2. Preliminary Results

The first author is grateful to NSERC (Canada) for supporting this project.

Clarke (1957) considered the M/M/1 queue, in which customers'arrival form a Poisson process with parameter λ, and service times of the customers are independent and identically exponentially distributed random variables with mean $1/\mu$. The ratio $\rho = \lambda/\mu$ is called the traffic intensity. The maximum likelihood estimates of the parameters λ and μ are given by

$$\hat{\lambda} = \frac{N_a}{t} \ , \qquad (2.1)$$

and

$$\hat{\mu} = \frac{N_s}{t_b} \ . \qquad (2.2)$$

Where, the system is observed for a fixed interval of time duration t. During this time period, there are N_a arrivals, N_s service completions and the service facility is busy for t_b time units.

Lilliefors (1966) has considered the problem of finding the confidence intervals for the actual M/M/1 traffic intensity given by

$$\hat{\rho} = \frac{\hat{\lambda}}{\hat{\mu}} \ . \qquad (2.3)$$

Thus , the traffic intensity is estimated by

$$\hat{\rho} = \frac{N_a/t}{N_s/t_b} \ . \qquad (2.4)$$

Consider the following ratio

$$\begin{aligned}\frac{\hat{\rho}}{\rho} &= \frac{(N_a/t)/(N_s/t_b)}{(\lambda/\mu)} \\ &= \frac{(2\mu t_b/2N_s)}{(2\lambda t/2N_a)} \ . \qquad (2.5)\end{aligned}$$

For large sample, Cox (1965) stated that $2\lambda t$ can be treated as a Chi-squared variate with $2N_a$ degrees of freedom and $2\mu t_b$ as a Chi-squared variate with $2N_s$ degrees of freedom. Thus, ρ/ρ has F-distribution with degrees of freedom $2N_s$ and $2N_a$. An appropriate probability statement at significance level α can be written as follows:

$$P\left[F_{1-\alpha/2}(2N_s,2N_a) \leq \frac{\hat{\rho}}{\rho} \leq F_{\alpha/2}(2N_s,2N_a)\right]$$
$$= 1-\alpha \ . \qquad (2.6)$$

Therefore, the upper and lower confidence limits for ρ are given by

$$\rho_u = \frac{\hat{\rho}}{F_{1-\alpha/2}(2N_s,2N_a)} \ , \qquad (2.7)$$

$$\rho_L = \frac{\hat{\rho}}{F_{\alpha/2}(2N_s,2N_a)} \ . \qquad (2.8)$$

If $f(\rho)$ is a monotonically increasing function of ρ, then the $100(1-\alpha)\%$ confidence intervals for $f(\rho)$ is

$$f(\rho_u) \leq f(\rho) \leq f(\rho_L) \ . \qquad (2.9)$$

3. Estimation Procedures for Priority Queues

Taylor and Karlin (1984) considered a single server queueing system with two types of customers so-called priority and non-priority. The customers arrive independently and formed a Poisson process with parameters α and β respectively. The customers' service times are independent and identically exponentially distributed with parameters γ and δ respectively. Service discipline is FCFS and the service of priority customers is never interrupted. A priority customer is allowed to enter the service immediately even if another nonpriority customer is already present in the service. The interrupted customer's service is resumed when there is no priority customer present in the system.

Notations

System arrival rate $= \lambda = \alpha + \beta$

Proportion of priority customers $= p = \alpha/\lambda$

Proportion of nonpriority customers $= q = \beta/\lambda$

The system mean service time is the approximately weighted means of the priority and nonpriority customers given by

$$\frac{1}{\mu} = \frac{1}{\lambda}\left[\frac{\alpha}{\gamma} + \frac{\beta}{\delta}\right] \qquad (3.1)$$

where μ is the system service rate.

The traffic intensity for the system, priority customers and nonpriority customers are given by

$$\rho = \frac{\lambda}{\mu} \qquad (3.2)$$

$$\Sigma = \frac{\alpha}{\gamma} \qquad (3.3)$$

and

$$\tau = \frac{\beta}{\delta}\,. \qquad (3.4)$$

It is clear from (3.1) that

$$\rho = \Sigma + \tau\,. \qquad (3.5)$$

Taylor and Karlin (1984) obtained the mean queue length for the priority and nonpriority customers in the steady state as follows:

$$L_p = \frac{\Sigma}{1-\Sigma}\,, \qquad (3.6)$$

and

$$L_n = \frac{\tau}{1-\Sigma-\tau}\left[1 + (\delta/\gamma)(\frac{\Sigma}{1-\Sigma})\right] .(3.7)$$

L_n is finite if the system traffic intensity $\rho(\rho = \Sigma + \tau)$ is less than 1.

For a simple M/M/1 queueing system with traffic intensity ρ, the mean queue length L is given by

$$L = \frac{\rho}{1-\rho} \qquad (3.8)$$

Suppose that the proportion p of the customers have priority and priority is independent of service time. Let $\delta = \gamma$, which implies $\Sigma = p\rho$ and $\tau = q\rho$. Then the expected queue length for the priority and nonpriority customers are given by

$$L_p = \frac{p\rho}{1-p\rho}\,, \qquad (3.9)$$

and

$$L_n = \frac{q\rho}{1-\rho}\left[1 + \frac{p\rho}{1-p\rho}\right] \qquad (3.10)$$

Therefore, the expected difference of queue length between nonpriority and priority customers is given by

$$D = L_n - L_p = \left[\frac{\rho}{1-p\rho}\right]\left[\frac{q-p+p\rho}{1-\rho}\right]. \qquad (3.11)$$

It can be shown that D is a monotonic increasing function of ρ $(0 < \rho < 1)$ as follows:

$$\frac{d}{d\rho}(lnD) = \left[\frac{1}{\rho} + \frac{p}{q-p+p\rho} + \frac{1}{1-\rho}\right].$$

The above is greater than zero, if $0 < \rho < 1$ and $0 \le p \le 0.5$. Hence, confidence limits of D can be written by substituting lower and upper limits of ρ using formulae (2.7) and (2.8).

4. Numerical Example

The estimate of mean queue length difference between nonpriority and priority customers by using equation (3.11) is given by

$$\hat{D} = \left[\frac{\hat{\rho}}{1-p\hat{\rho}}\right]\left[\frac{q-p+p\hat{\rho}}{1-\hat{\rho}}\right], \qquad (4.1)$$

where $\hat{\rho}$ is the estimated parameter of traffic intensity for the system.

The upper and lower confidence limits for D are given by

$$D_u = \left[\frac{\hat{\rho}_u}{1-p\hat{\rho}_u}\right]\left[\frac{q-p+p\hat{\rho}_u}{1-\hat{\rho}_u}\right], (4.2)$$

and

$$D_L = \left[\frac{\hat{\rho}_L}{1-p\hat{\rho}_l}\right]\left[\frac{q-p+p\hat{\rho}_L}{1-\hat{\rho}_L}\right], \quad (4.3)$$

where $\hat{\rho}_u$ and $\hat{\rho}_L$ are computed using formulae (2.7) and (2.8) respectively. Tables 4.1, 4.2 and 4.3 compute the width of 90% confidence intervals with various number of arrivals and service of customers at $p = 0.2$. Similarly, Tables 4.4, 4.5 and 4.6 present the corresponding results computed at $p = 0.4$.

Table 4.1. 90% confidence intervals for various values of arrivals and services completions with traffic intensity $\rho = 0.2$ and $D = 0.1667$ at $p = 0.2$.

$n_a = n_s$	D_u	D_L	Width of CI
20	0.3656	0.0854	0.2802
30	0.3105	0.0969	0.2136
40	0.2853	0.1032	0.1821
50	0.2673	0.1088	0.1585
60	0.2557	0.1127	0.1430

Table 4.2 90% confidence intervals for various values of arrivals and service completions with traffic intensity $\rho = 0.4$ and $D = 0.4927$ at $p = 0.2$.

$n_a = n_s$	D_u	D_L	Width of CI
20	1.7737	0.2095	1.5642
30	1.2983	0.2430	1.0553
40	1.1185	0.2642	0.8543
50	1.0020	0.2819	0.7201
60	0.9318	0.2945	0.6373

Table 4.3. 90% confidence intervals for various values of arrivals and service completions with traffic intensity $\rho = 0.5$ and $D = 0.7778$ at $p = 0.2$.

$n_a = n_s$	D_u	D_L	Width of CI
20	5.0449	0.2944	4.7505
30	2.8940	0.3456	2.5484
40	2.2792	0.3782	1.9191
50	1.9558	0.4067	1.5491
60	1.7648	0.4283	1.3365

Table 4.4. 90% confidence intervals for various values of arrivals and service completions with traffic intensity $\rho = 0.2$ and $D = 0.0761$ at $p = 0.4$.

$n_a = n_s$	D_u	D_L	Width of CI
20	1.1759	0.0347	1.1412
30	0.1619	0.0402	0.1217
40	0.1460	0.0432	0.1028
50	0.1348	0.0459	0.0889
60	0.1277	0.0479	0.0798

Table 4.5. 90% confidence intervals for various values of arrivals and service completions with traffic intensity $\rho = 0.4$ and $D = 0.2857$ at $p = 0.4$.

$n_a = n_s$	D_u	D_L	Width of CI
20	1.3452	0.1009	1.2443
30	0.9290	0.1200	0.8090
40	0.7768	0.1329	0.6439
50	0.6802	0.1439	0.5363
60	0.6229	0.1521	0.4708

Table 4.6. 90% confidence intervals for various values of arrivals and service completions with traffic intensity $\rho = 0.5$ and $D = 0.5$ at $p = 0.4$.

$n_a = n_s$	D_u	D_L	Width of CI
20	4.4305	0.1518	4.2787
30	2.3735	0.1847	2.1888
40	1.8195	0.2063	1.1613
50	1.5086	0.2256	1.2830
60	1.3372	0.2405	1.0967

Concluding Remarks

The width of 90% confidence intervals for the expected difference of queue length for the nonpriority and priority customers in M/M/1 queueing system are computed. Tables 4.1 to 4.6 indicate that the width of confidence interval decreases as the number of arrival and service of the customers increases. Obviously, one can observe easily from Tables 4.3 and 4.6 that the confidence interval increases rapidly when the traffic intensity increases. There is a possibility that ρ_u could be greater than one when formula (2.7) is used for computing ρ_u. Under such circumstances, the statistical techniques have limitations. The cautious approach is required for estimating the parameter.

References

Allen, A. O. *Probability, Statistics and Queueing Theory with Computer Science Applications. (1978). Academic Press, New York.*

Bhat, U. N. and Rao, S. S. Statistical analysis of queueing systems. *QUESTA, I, (1987), 217-247.*

Clarke, A. B. Maximum likelihood estimates in a simple queue. *Ann. Math. Statist., 28, (1957), 1036-1040.*

Cox, D. R. *Some problems of statistical analysis connected with congestion. Proc. Symp. on Congestion Theory. (eds. W. L. Smith and W. B. Wilkinson), (1965) University of North Carolina, Chapel Hill, N. C.*

Gross, D. and Harris, C. M. *Fundamentals of Queueing Theory. Second edition. (1985), Wiley, New York.*

Jain, S. Comparison of confidence intervals of traffic intensity for $M/E_k/1$ queueing systems. *Statistical Hefte, 32, (1991), 167-174.*

Lilliefors, H. W. Some confidence intervals for queues. *Oper. Res., 14, (1966), 723-727.*

Taylor, H. M. and Karlin, S. *An Introduction to Stochastic Modeling. (1984) Academic Press, New York.*

Using both Symbolic and Classical methods to analyse complex data set with the SAS system.

J.L. BLANCHARD*
M. GETTLER SUMMA**

* EDF DER, 1 av du général de Gaulle, 92141 Clamart Cedex, France
mail : jean-louis.blanchard@der.edf.fr
** université PARIS IX Dauphine, Lise-Ceremade, 1 place du ml de Lattre de Tassigny, 75016 Paris
email : summa@etud.dauphine.fr

Our developments are about a study on the representation and on the analysis of ergonomists knowledge. The context of this study is the evaluation of a control-room for nuclear power plants.

It includes the creation and the adaptation of the theoretical framework chosen to resolve the problem (Symbolic Data Analysis) and a realization of a prototype coupling numerical algorithms with symbolic methods using the SAS software.

I. Introduction

EDF (French Electricity National Company) is testing a new type of control-room with computer-based interface.

Ergonomists have to evaluate this new control-room. During trials, operators who drive a simulated nuclear power plant are observed by Ergonomists. The latters note down the operator's behaviour and they key in the principal actions (from the ergonomic point of view) : moving, grouping, speaking, etc.

The evaluation of the control-room requires to have a global approach of the operator's operation methods :

1. Activity of each operator
2. Activity of the team

II. Methods and algorithms

1) Knowledge representation

The heterogeneousness of data (operators have different tasks, described with different variables), and the studied themes (for example the notion of "activity" in a team), have imposed us to consider the problem of knowledge representation and computing, in the large framework of Symbolic Data Analysis [Diday 93]. The activity of a team can be defined by using the mathematical form of the synthetic objects :

```
Activity1 :[duration=[0h25, 1h10[
        ^ [glance(operator1)={0.9 screen, 0.1 synoptic}]
        ^ [glance(operator2)={0.8 screen, 0.2 elsewhere}]
...
```

2) Methods

The different proposed analysis for the studied themes have been conceived using both numerical classical methods and symbolic methods. A such idea is already used in the generalization of symbolic objects coming from machine-learning [Summa 93].

3) Algorithms

In mind to tackle complex data structures, we couple a classical statistical software (SAS) with symbolic methods. To goal the different activities, we develop a statistical toolbox.

The symbolic methods developed in the toolbox let us use:
- variables with several levels in the same time (uncertainties on values)
- links between variables (for example : if "groupment"=no then "with who" has no sense)

The toolbox can be decomposed in two parts :

Symbolic methods
- symbolic histograms
- symbolic hierarchical clustering
- symbolic pyramidal clustering (with base constrained)
- symbolic explanation of clusters

Transformations methods
- dissimilarity computing
- probability computing

and the toolbox lets us use all classical methods already available in the SAS System.

III. Application to the S3C project

The interest of the coupling is the facility to toggle between classical methods and symbolic methods.

A good example of this possibility is in the research of a plane representation of different observations and in the research of the operator's activity trajectories. These activities are described by the regroupment with the other operators in the control-room.

Initial data describe the regroupment of the operator during regular time intervals (one minute).

The methods used are a sequence of classical methods and symbolic methods.

For the research of plane representation :

1) Symbolic dissimilarity computing to transform data in numerical form

then 2) Classical Multidimensionnal Scaling to represent data in a plane

then 3) Classical K-means clustering

then 4) Symbolic explanation of the cluster.

For the research of the operator's activity trajectories :

1) Symbolic computing of probability to transform data in a numerical table

2) Classical Factorial analysis on the numerical table

3) Compute the trajectories as supplementary individuals

4) Symbolic explanation of the axes

IV. Conclusion

The union of classical data analysis methods with symbolic data analysis methods allows to use complex data sets (uncertainties on values, links between variables) keeping the possibility of using classical data analysis.

The results might be encouraging (the real data have not been received yet) but the development of symbolic methods with a classical statistical software as the SAS system is very heavy (the basic structure of SAS data set is made with row-columns tables).

Coupling an Object-Oriented Database with a classical statistical software using a language like C++ seems more appropriate

V. Bibliography

Brito P., "Comparing Numerical and Symbolic Clustering : Application to Pseudoperiodic Light Curves of Stars" in Computational Statistics, Physica-Verlag, 1992.

Carvalho F.A.T., "Description of a knowledge base of symbolic objects by histograms" in Computational Statistics, Physica-Verlag, 1992.

Diday E., "From Data to Knowledge : Probabilist Objects for a Symbolic Data Analysis", in Computational Statistics, Physica-Verlag, 1992.

Diday E., "An Introduction to Symbolic Data Analysis", Research report INRIA n° 1936, Rocquencourt, 1993.

Diday E., "Quelques aspects de l'analyse des donnees symboliques", Research report INRIA n 1937, Rocquencourt, 1993.

Gettler Summa M., Ferraris J., Perinel E., "Automatic aid to symbolic interpretation in clustering" in IFCS conferences, Paris, 1993

Gower J.C., "Measures of similarity, dissimilarity, and distance" in Encyclopedia of Statistical Sciences, Wiley, New York, 1988

Hatabian G., "La statistique au service de l'ergonomie dans la conduite des centrales nucleaires", Internal report EDF n HI-23/6568, Clamart, 1989.

Ho Tu Bao, Diday E., Gettler Summa M., "Generating rules for expert system from observations" in Pattern Recognition Letters 7 pp265-271, North Holland, 1988.

Jobson J.D., "Applied Multivariate data analysis", volume II, Springer Verlag, New York, 1992.

Saporta G., "Probabilites, Analyse des donnees, et Statistiques", Technip, Paris, 1990

Touati M., "Synthese d'objets, application a l'orientation des etudiants de l'universite d'Alger", DEA Report, Paris Dauphine University, 1993

VI. Slides

Symbolic Data Analysis

- **1 st feature :**

Individuals are described by logical expressions
Cepe : [HatColor = { red, green }]
^ [FootHeight={ small }]
^ ...

The description can be complex
Boletus : [HatColor = { yellow, brown }]
^ [Height = [0 , 7] if HatColor=yellow,
[7 , 15] if HatColor=brown]
^ ...

and can use external knowledge
if HatPresence=no then HatColor has no sense
if HatColor=black then Smell is nauseous or pleasant

The mushroom data set

Mush.	Green House	Begin	End	Hat Presence	Hat Shape	Hat Color	Foot Height	Smell
cepe	Pignac	1	2	p	c, s	r, g	s	n, p
agaric	Pignac	1	2	p	s, t	g, w	s, m	o, p
chanterelle	Pignac	1	2	p	s	g	s	n
cepe	Pignac	2	3	p	s, t	r, g	t, m	p
agaric	Pignac	2	3	p	s, t	w, b	m, t	n, p
chanterelle	Pignac	2	3	p	s	g	s	n
cepe	Pignac	3	4	p	t	w, g	m	p
agaric	Pignac	3	4	p	t, s	g, b	s, m	o, n
chanterelle	Pignac	3	4	p	c	y	m	n
cepe	Pignac	4	5	a	NS	NS	m	p, o
agaric	Pignac	4	5	a	NS	NS	m, t	n, p
chanterelle	Pignac	4	5	a	NS	NS	t	o
cepe	Pignac	5	6	a	NS	NS	m, t	p, o
agaric	Pignac	5	6	a	NS	NS	m, t	o, p
chanterelle	Pignac	5	6	a	NS	NS	s	o
cepe	Tropic	1	2	p	t, s, c	g, w, y, b	m, s	o, n, p
agaric	Tropic	1	2	p, a	t, c, NS	b, g, NS	m, t	n, p
chanterelle	Tropic	1	2	p	t	b	s	n
cepe	Tropic	2	3	p	s, c, t	w	s, t, m	p, n
agaric	Tropic	2	3	p, a	s, t, c, NS	g, r, NS	s, t	n, p
...	...	...	...	...	...	...	...	...

HC = { r, g } : during the day, Hat Color has been either red, or green.

SDA (continued)

- **2 nd feature :**

INDIVIDUALS
CONCEPTS } **are described in the same syntax**

boletus : [HatColor = { white }]
^ [Height = [1 , 3]]

meadow mushrooms : [HatColor = { white }]
^ [Height = [1 , 3]]

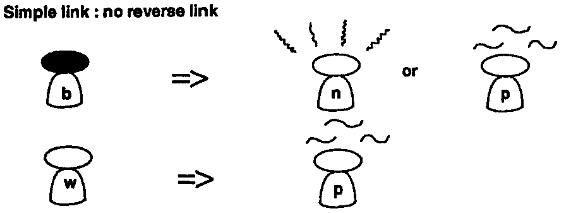
Links between variables

Strong link : the reverse link exists

HatPresence = absent => HatColor and HatShape have no sense

HatColor has no sense or HatShape has no sense => HatPresence = absent

Simple link : no reverse link



SDA (continued)

- **3 rd feature :**

Several ways to describe individuals or concepts

Intension

meadow mushrooms : [HatColor = { white }]
^ [Height = [1 , 3]]

Extension

among the set of initial individuals Ω

ext_{Ω}(meadow) : {chanterelle, agaric}

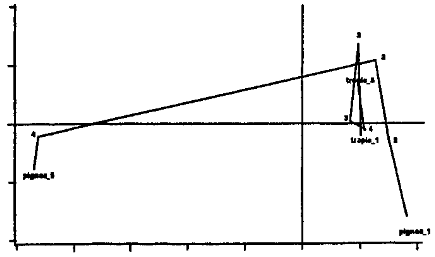
among the set of possible descriptions Θ

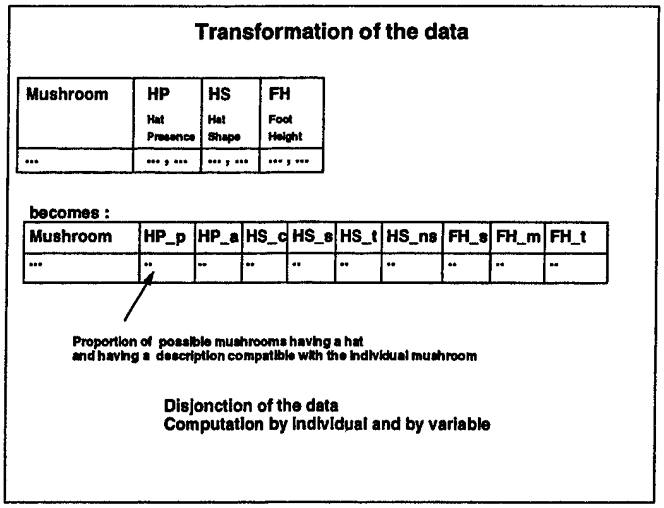
ext_{Θ}(cepe) = {(circle, red), (circle, green), (square, red) (square, green) }

Study of the mushroom evolution

- **Trajectories of the green house evolutions on the first factorial plane (PCA)**
  - **» transformation of the data in a pseudo-disjunctive form**
  - **» principal component analysis**
  - **» trajectory drawings**

The green house evolution trajectories





Transformation of the data

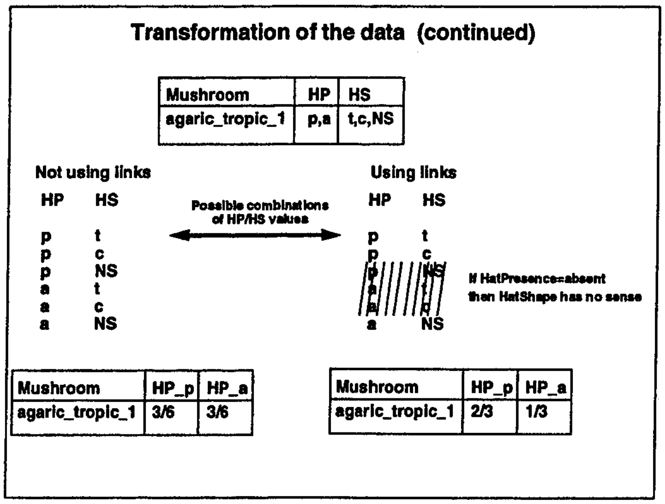
Mushroom	HP Hat Presence	HS Hat Shape	FH Foot Height
...	... ; ...	... ; ...	... ; ...

becomes :

Mushroom	HP_p	HP_a	HS_c	HS_s	HS_t	HS_ns	FH_s	FH_m	FH_t
...	..	..	..	..	..	..	..	..	..

Proportion of possible mushrooms having a hat
and having a description compatible with the individual mushroom

Disjonction of the data
Computation by individual and by variable



Transformation of the data (continued)

Mushroom	HP	HS
agaric_tropic_1	p,a	t,c,NS

Not using links

HP	HS
p	t
p	c
p	NS
a	t
a	c
a	NS

Possible combinations of HP/HS values

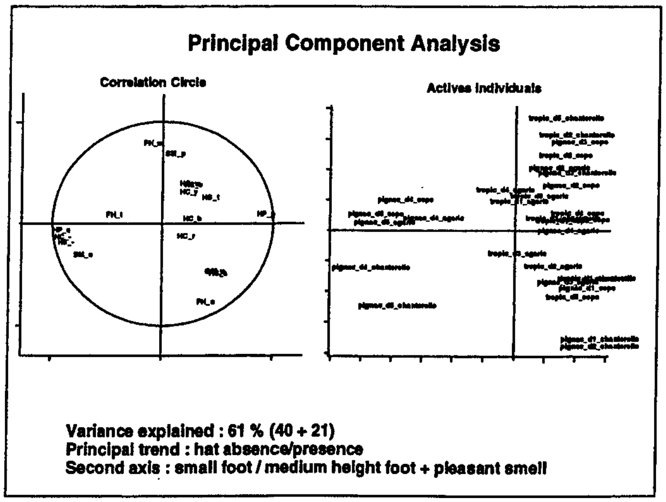
Using links

HP	HS
p	t
p	c
~~p~~	~~NS~~
~~a~~	~~t~~
~~a~~	~~c~~
a	NS

If HatPresence=absent then HatShape has no sense

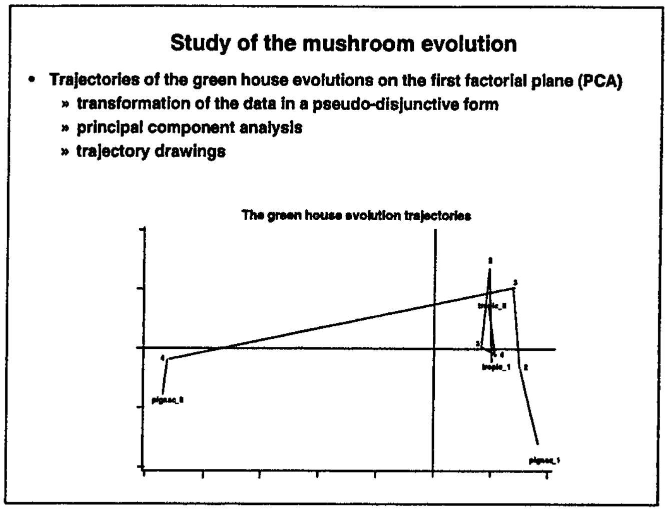
Mushroom	HP_p	HP_a
agaric_tropic_1	3/6	3/6

Mushroom	HP_p	HP_a
agaric_tropic_1	2/3	1/3



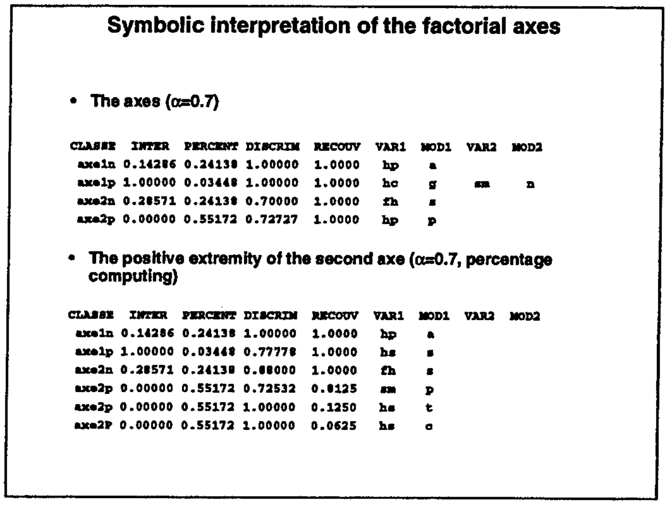
Principal Component Analysis

Variance explained : 61 % (40 + 21)
Principal trend : hat absence/presence
Second axis : small foot / medium height foot + pleasant smell



Study of the mushroom evolution

- Trajectories of the green house evolutions on the first factorial plane (PCA)
  - » transformation of the data in a pseudo-disjunctive form
  - » principal component analysis
  - » trajectory drawings



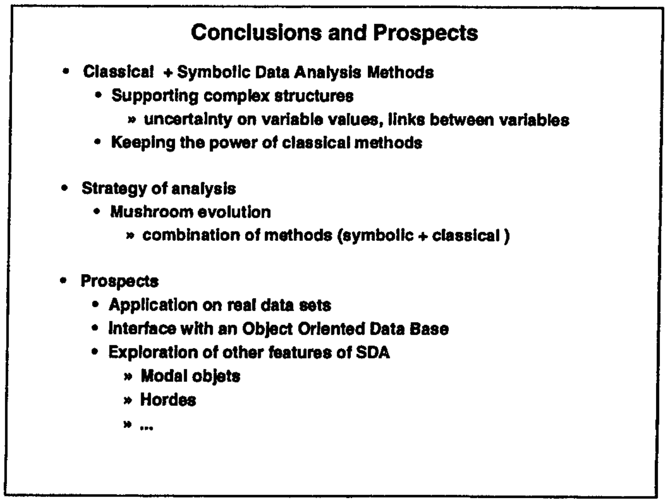
Symbolic interpretation of the factorial axes

- The axes (α=0.7)

```
CLASSE  INTER   PERCENT DISCRIM  RECOUV  VAR1  MOD1  VAR2  MOD2
 axe1n 0.14286 0.24138 1.00000  1.0000   hp    a
 axe1p 1.00000 0.03448 1.00000  1.0000   hc    g     sm    n
 axe2n 0.28571 0.24138 0.70000  1.0000   fh    s
 axe2p 0.00000 0.55172 0.72727  1.0000   hp    p
```

- The positive extremity of the second axe (α=0.7, percentage computing)

```
CLASSE  INTER   PERCENT DISCRIM  RECOUV  VAR1  MOD1  VAR2  MOD2
 axe1n 0.14286 0.24138 1.00000  1.0000   hp    a
 axe1p 1.00000 0.03448 0.77778  1.0000   hs    s
 axe2n 0.28571 0.24138 0.88000  1.0000   fh    s
 axe2p 0.00000 0.55172 0.72532  0.8125   sm    p
 axe2p 0.00000 0.55172 1.00000  0.1250   hs    t
 axe2P 0.00000 0.55172 1.00000  0.0625   hs    c
```



Conclusions and Prospects

- Classical + Symbolic Data Analysis Methods
  - Supporting complex structures
    - » uncertainty on variable values, links between variables
  - Keeping the power of classical methods
- Strategy of analysis
  - Mushroom evolution
    - » combination of methods (symbolic + classical)
- Prospects
  - Application on real data sets
  - Interface with an Object Oriented Data Base
  - Exploration of other features of SDA
    - » Modal objets
    - » Hordes
    - » ...

Small Sample Conditional Inference in Biostatistics

John E. Kolassa*

Department of Biostatistics, University of Rochester Medical Center, Rochester, NY 14642 USA

Abstract

Kolassa and Tanner (1994) present the Gibbs–Skovgaard algorithm for approximate conditional inference. This algorithm makes use of the double saddlepoint approximation to the conditional distribution function of a sufficient statistic given the remaining sufficient statistics. This approximation is used with the Gibbs Sampler to generate a Markov chain. The equilibrium distribution of this chain approximates the joint distribution of the sufficient statistics associated with the parameters of interest conditional on the observed values of the sufficient statistics associated with the nuisance parameters. In this paper recent extensions to this methodology are recounted, and open questions related to the existence and accuracy of the resulting approximation to the desired distribution are discussed.

1. Introduction

Kolassa and Tanner (1994) construct an algorithm for simulating observations from distributions approximating null conditional distributions in generalized linear models, in order to construct conditional significance tests. The suggest using the Gibbs sampler to construct a Markov Chain whose null distribution is the conditional distribution of interest, and approximating this chain by sampling from the double saddlepoint conditional cumulative distribution function approximation of Skovgaard (1987) instead of from the true conditional cumulative distribution functions. This approximation depends on a parameter m roughly measuring the number of independent and identically distributed observations represented in the data set. Besag and Clifford (1989, 1991) discuss methods by which such a Markov chain may be used for frequentist inference. This paper surveys work extending that of Kolassa and Tanner (1994) in a number of ways. Theorems are cited governing irreducibility and ergodicity of the constructed Markov chain. Accuracy of the resulting equilibrium distribution as an approximation to the desired distribution, the use of a higher–order approximation of Kolassa (1992a), and the extension of Kolassa (1992b) to cases in which the saddlepoint is not defined, are all discussed.

Tierney (1991) reviews Markov chain convergence results in the more general case in which Hilbert space techniques are inapplicable; this paper makes use of such methods. These methods are similar to those used by Roberts and Polson (1994), but are extended to the case where the sampling performed is only approximately according to the Gibbs scheme, using the double saddlepoint approximation. Roberts and Smith (1994) discuss conditions of aperiodicity and irreducibility necessary for convergence. These questions are considered in this paper.

This paper is organized as follows. First, Markov chain terminology, Gibbs sampling, and the double saddlepoint distribution function approximation are reviewed. An example of the method of Kolassa and Tanner (1994) is recounted. Irreducibility of the their Markov chain is considered in the setting of regression problems. Results are recounted showing geometric convergence to an equilibrium distribution, dependent on m. A simple example is given demonstrating that stronger convergence is not in general possible. The equilibrium distribution is conjectured to converge to the target distribution as m increases.

2. Markov Chain Terminology

The methods used in this paper to prove convergence of the constructed Markov chains are similar to those used by other authors. To make connections between this work and other Markov chain literature clearer, some common definitions concerning Markov chains are introduced. The first defines the structure of transitions from one step in the chain to another. The second considers whether the state space may be divided into two spaces, between which the chain never travels. If this is the case, there are an infinite number of equilibrium distributions for the chain, depending on how much mass is initially allocated to each subspace. The third definition concerns whether the measure induced by certain transitions in the chain can be bounded below by a measure that does not depend on

* Supported by grant CA 63050 from the National Institutes of Health

the initial transition. Nummelin (1984) presents many Markov chain convergence results depending on this property.

Definition 2.1 : Suppose transitions in a Markov chain $(\boldsymbol{T}^{(n)})$ with state space $\mathfrak{T}$ from state $\boldsymbol{y}$ are given by the density $P(\boldsymbol{y}, \boldsymbol{t})$ with respect to a measure μ and Borel sets $\mathcal{T}$ relative to the relevant topology on $\mathfrak{T}$. Let $P(\boldsymbol{y}, \cdot)$ be the associated measure on $\mathfrak{T}$, and recursively define the measures $P^{(1)}(\boldsymbol{y}, \cdot) = P(\boldsymbol{y}, \cdot)$, and $P^{(n)}(\boldsymbol{y}, \cdot) = \int_{\mathfrak{T}} P(\boldsymbol{y}, d\boldsymbol{z}) P^{(n-1)}(\boldsymbol{z}, \cdot)$.

Definition 2.2 : For any measure μ on $\mathcal{T}$, the chain $(\boldsymbol{T}^{(n)})$ is μ–irreducible if $P[\exists n \ni \boldsymbol{T}^{(n)} \in A | \boldsymbol{T}_0 = \boldsymbol{t}] > 0$ for all $\boldsymbol{t} \in \mathfrak{T}$ and for all $A \in \mathcal{T} \ni \mu(A) > 0$. Equivalently, the chain is irreducible if $\mu(\{\boldsymbol{y} | \sum_{m=0}^{\infty} P^{(m)}(\boldsymbol{t}, \boldsymbol{y}) = 0\}) = 0 \; \forall \boldsymbol{t} \in \mathfrak{T}$.

Definition 2.3 : A set $C \subset \mathfrak{T}$ is small if and only if there exists a constant $\alpha > 0$, and a probability measure ν on $\mathfrak{T}$ such that $P(\boldsymbol{t}, \cdot) \geq \alpha\nu(\cdot)$ for $\boldsymbol{t} \in C$.

Nummelin (1984) describes several forms of convergence results for Markov chains. Two are considered here:

Definition 2.4 : A Markov chain is uniform ergodic, if there exists a probability measure π, a constant $r \in (0, 1)$, and a constant M such that $\|P^{(n)}(\boldsymbol{y}, \cdot) - \pi(\cdot)\|_{TV} < Mr^n$, and is geometrically ergodic, if which there exists a probability measure π, a constant $r \in (0, 1)$, and a function $M(\boldsymbol{y})$ such that $\|P^{(n)}(\boldsymbol{y}, \cdot) - \pi(\cdot)\|_{TV} < M(\boldsymbol{y})r^n$. Here $\| \cdot \|_{TV}$ is the total variation norm on the space of finite measures on $\mathfrak{U}$.

Definition 2.5 : A Markov chain has period q if there exist disjoint measurable subsets $\mathcal{T}_0, \ldots, \mathcal{T}_{q-1}$ of $\mathfrak{T}$ such that $E[\boldsymbol{T}^{(n)} \in \mathcal{T}_j^c | \boldsymbol{T}^{(n-1)} = \boldsymbol{t}] = 0$ whenever $\boldsymbol{t} \in \mathcal{T}_i$ and $i = (j-1) \bmod q$, and if q is the largest integer having this property.

3. Gibbs Sampling

The Gibbs sampler is a popular Markov chain method useful for yielding a sample from a posterior or likelihood density. It was first introduced by Geman and Geman (1984) in the context of image reconstruction. The data augmentation algorithm of Tanner and Wong (1987), introduced as a device for the calculation of posterior distributions, is a 'two-component' version of the Gibbs sampler. See Tanner (1993) for background details and important references.

Let the symbol $p(\cdots | \cdots)$ denote the distribution of those random variables listed before the vertical line conditional on those listed after, and let the vector $\boldsymbol{T}_{-j}$ denote the vector $\boldsymbol{T}$ with component j deleted. To obtain a sample from the joint conditional distribution $p(T_1, \cdots, T_a | T_{a+1}, \ldots, T_d)$, the systematic scan Gibbs sampler iterates the following loop: Sample

1) $T_1^{(n)}$ from $p(T_1 | T_2^{(n-1)}, \cdots, T_a^{(n-1)}, T_{a+1}, \ldots, T_d)$.
2) $T_2^{(n)}$ from $p(T_2 | T_1^{(n)}, T_3^{(n-1)}, \cdots, T_a^{(n-1)}, T_{a+1}, \ldots, T_d)$.

$\vdots$

a) $T_a^{(n)}$ from $p(T_a | T_1^{(n)}, \cdots, T_{a-1}^{(n)}, T_{a+1}, \ldots, T_d)$.

If the algorithm converges, for a sufficiently large value of n we can take $T_1^{(n)}, \cdots, T_a^{(n)}$ as a simulated observation from the equilibrium distribution $p(T_1, \cdots, T_a | T_{a+1}, \ldots, T_d)$ of the Markov chain. Independently replicating this Markov chain l times produces an independent and identically distributed sample of size l from the distribution of interest.

4. Double Saddlepoint Approximation

Often the one–dimensional marginal distributions required for Gibbs sampling are unavailable. Kolassa and Tanner (1994) suggest instead sampling from double saddlepoint approximations to the appropriate conditional cumulative distribution functions. The double saddlepoint cumulative distribution function approximation of Skovgaard (1987) generalizes the secant approximation due to Lugannani and Rice (1980) examined by Skates (1993). Suppose a vector $\boldsymbol{T}$ arises as the mean of m independent and identically distributed random vectors, each with cumulant generating function K, and one wishes to approximate the distribution of T_u conditional on the value of $\boldsymbol{T}_{-u} = (T_1, \cdots, T_{u-1}, T_{u+1}, \cdots, T_d)$. In the context above this will be applied for $u \leq a$. The double saddlepoint approximation involves solving the multivariate saddlepoint equations both for the full distribution of $\boldsymbol{T}$ and for the distribution of the shorter random vector $\boldsymbol{T}_{-u}$. The approximate conditional distribution function is

$$\Phi(\sqrt{m}\hat{w}_1) + m^{-1/2}\,\phi(\sqrt{m}\hat{w}_1)\left(\frac{1}{\hat{w}_1} - \frac{1}{\hat{z}}\right) + O(m^{-3/2}), \tag{1}$$

where

$$\hat{z} = \hat{\beta}_1 \sqrt{\left|K''(\hat{\boldsymbol{\beta}})\right|} / \sqrt{\left|K''_{-u}(\tilde{\boldsymbol{\beta}})\right|}$$

$$\hat{w}_1 = \operatorname{sgn}(\hat{\beta}_u)\sqrt{2[(\hat{\boldsymbol{\beta}}^T - \tilde{\boldsymbol{\beta}}^T)K'(\hat{\boldsymbol{\beta}}) - K(\hat{\boldsymbol{\beta}}) + K(\tilde{\boldsymbol{\beta}})]}, \tag{2}$$

and $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\beta}}$ solve

$$K^j(\hat{\boldsymbol{\beta}}) = t^j \; \forall j \text{ and } K^j(\tilde{\boldsymbol{\beta}}) = t^j \; \forall j \neq u, \; \tilde{\beta}_u = 0, \tag{3}$$

K''_{-u} is the $(d-1)\times(d-1)$ submatrix of second derivative matrix of K, corresponding to all components of $\boldsymbol{\beta}$ and $\boldsymbol{T}$ except the first, and Φ and ϕ are the normal distribution function and density respectively. The vectors $\hat{\boldsymbol{\beta}}$ and $\tilde{\boldsymbol{\beta}}$ represent maximum likelihood estimators for the canonical exponential family containing $\boldsymbol{T}$.

Also of interest are inversion techniques for lattice distributions. Skovgaard (1987) derives a counterpart of (1) in the lattice case, in which $\hat{\beta}_u$ is replaced by $2\sinh(\frac{1}{2}\hat{\beta}_u)$ in the definition of $\hat{z}$, and in which t^u is corrected for continuity when calculating $\hat{\boldsymbol{\beta}}$. That is, if possible values for T_u are Δ units apart, $\hat{\boldsymbol{\beta}}$ solves $K'(\hat{\boldsymbol{\beta}}) = \tilde{\boldsymbol{t}}$ where $\tilde{t}^j = t^j$ if $j \neq u$ and $\tilde{t}^u = t^u - \frac{1}{2}\Delta$.

Later results concerning using the double saddlepoint approximation in conjunction with Gibbs sampling will require that the approximation be monotone. The derivative of (1) is

$$\sqrt{m}\phi(\sqrt{m}\,\hat{w}_1)\frac{\hat{w}_1}{\hat{z}}\left(1+\frac{(d\hat{z}/d\hat{w}_1)\hat{z}^{-1}\hat{w}_1^{-1}-\hat{z}\hat{w}_1^{-3}}{m}\right)\frac{\partial\hat{w}_1}{\partial t^u}; \tag{4}$$

this constitutes an approximate conditional density for T_u in the continuous case, and in the lattice case probability atoms are integrals of (4). Then (1) is a distribution function for

$$m > \max(\hat{z}\hat{w}_1^{-3} - (d\hat{z}/d\hat{w}_1)\hat{z}^{-1}\hat{w}_1^{-1}). \tag{5}$$

Skates (1993) discusses monotonicity of similar distribution function approximations.

We evaluate the monotonicity criterion, (5). Note that $\partial\hat{z}/\partial t^u = \hat{z}\left(\hat{\beta}_l^u/\hat{\beta}_l + \frac{1}{2}\hat{\kappa}_{ij}\hat{\kappa}^{ijl}\hat{\beta}_l^u\right)$, and

$$\begin{aligned}\partial\hat{w}_1/\partial t^u &= \hat{w}_1^{-1}\hat{\beta}_i^u\left(\hat{\kappa}^i + (\hat{\beta}_j - \tilde{\beta}_j)\hat{\kappa}^{ij} - \hat{\kappa}^i\right)\\ &= \hat{w}_1^{-1}\hat{\beta}_i^u(\hat{\beta}_j - \tilde{\beta}_j)\hat{\kappa}^{ij}.\end{aligned}$$

Also, $K^i(\hat{\boldsymbol{\beta}}) = t^i$, $\hat{\kappa}^{ij}\hat{\beta}_j^u = \delta^{iu}$, and $\hat{\beta}_l^u = \hat{\kappa}_{ul}$. Then $\partial\hat{z}/\partial t^u = \hat{z}\left(\hat{\kappa}_{uu}/\hat{\beta}_u + \frac{1}{2}\hat{\kappa}_{ij}\hat{\kappa}^{ijl}\hat{\kappa}_{ul}\right)$, and $\partial\hat{w}_1/\partial t^u = \hat{w}_1^{-1}(\hat{\beta}_u - \tilde{\beta}_u) = \hat{w}^{-1}\hat{\beta}_u$, since $\tilde{\beta}_u = 0$. Hence $d\hat{z}/d\hat{w}_1 = \hat{z}\hat{w}_1(\hat{\kappa}_{uu}/\hat{\beta}_u + \frac{1}{2}\hat{\kappa}_{ij}\hat{\kappa}^{ijl}\hat{\kappa}_{ul})/\hat{\beta}_u$. Furthermore, the density associated with (1) can be expressed as

$$\begin{aligned}&\sqrt{m}\,\phi(\sqrt{m}\,\hat{w}_1)\sqrt{\left|K''(\hat{\boldsymbol{\beta}})\right| / \left|K''_{-u}(\tilde{\boldsymbol{\beta}})\right|}\\ &\quad\times\left(1+\frac{(d\hat{z}/d\hat{w}_1)\hat{z}^{-1}\hat{w}_1^{-1}-\hat{z}\hat{w}_1^{-3}}{m}\right).\end{aligned}$$

Terms of order $O(\sqrt{m})$ comprise the double saddlepoint density approximation of Barndorff-Nielsen and Cox (1979). For further discussion and references see Kolassa (1994a).

5. An Example

Kolassa and Tanner (1994) apply Gibbs-Skovgaard approach to higher-way contingency tables. Consider the distribution of elements in $d_1 \times d_2 \times d_3$ contingency tables, expressed as X_{ijk} where $i \in \{1,\cdots,d_1\}$, $j \in \{1,\cdots,d_2\}$, and $k \in \{1,\cdots,d_3\}$, conditional on one dimensional marginals. Express the table in terms of $d_1d_2d_3$ sufficient statistics, of which 1 is the overall total, $d_1 - 1$ are first unidimensional totals, $d_2 - 1$ are second unidimensional marginal totals, and $d_3 - 1$ are third unidimensional marginal totals, $(d_1-1)(d_2-1)$ are first bidimensional totals, $(d_1-1)(d_3-1)$ are second bidimensional totals, $(d_2-1)(d_3-1)$ are third bidimensional totals, and the remaining $(d_1-1)(d_2-1)(d_3-1)$ sufficient statistics are the entries with none of their indices at the highest values. The first $d_1 + d_2 + d_3 - 2$ sufficient statistics are ancillary to the null hypothesis of complete independence nested within the saturated model for Poisson means. Other sufficient statistics are ancillary and are conditioned on. Consider the following data describing the presence or absence of torus mandibularis among male and female Inuits aged 41-50 in three different groups, collected by Muller and Mayhall (1971), and cited by Bishop, Fienberg, and Holland (1975):

Sex	Igloolik Group		Hall Beach Group		Aleut Group	
	Pres.	Abs.	Pres.	Abs.	Pres.	Abs.
M	10	0	4	2	4	5
F	6	4	4	0	2	2

These data were chosen to assess the quality of the Markov chain algorithm in a situation in which usual asymptotic approximations may be inappropriate. The hypothesis of independence was tested by generating random tables using the Gibbs sampler and Skovgaard's approximation as described above. Statistics for Pearson's χ^2 Test and the Likelihood Ratio Test were calculated for each simulated table.

We simulated 5,000 independent Markov chains for 200 iterations. For each integer n between 1 and 200 we estimated the p-value after iteration n generated by each test statistic by calculating the test statistic for the observed table, and for each of the simulated tables represented by the state of the chain at time n. We report as the estimated p-value the proportion of sample tables with a test statistic value as high or higher than the observed value. Convergence was assessed by observing how these estimates change with n; in this example after fewer than 25 iterations the estimated p-values become stable. The asymptotic p-values based on the Pearson and likelihood ratio statistics are 0.004 and 0.001, respectively. The corresponding values for the Gibbs-Skovgaard algorithm are 0.136 and 0.063.

6. Irreducibility of Chains for Regression Models

Kolassa (1994b) considers certain regression models, and determines when the Gibbs sampling Markov chain applied to the sufficient statistics

$$\boldsymbol{T} = \boldsymbol{Z}^T\boldsymbol{Y}, \text{ with } \boldsymbol{Y} \in \prod_j \mathfrak{Y}_j \tag{6}$$

is irreducible. The continuous case result is straight–forward:

Theorem 6.1 : For the statistics $\boldsymbol{T}$ of (6), where each $\mathfrak{Y}_j$ is a connected open subset of $\mathbb{R}$, and if a formal Gibbs sampling scheme which, when sampling component u conditional on $\boldsymbol{t}^0_{-u}$, samples from a distribution on $\{t_u | \boldsymbol{t} \in \mathfrak{T},\ \boldsymbol{t}_{-u} = \boldsymbol{t}^0_{-u}\}$ having a positive density with respect to Lebesgue measure, then the chain is irreducible.

The following example shows that the discrete case is more delicate. Suppose that $\mathfrak{Y}_u$ is a subset of the integers from 1 to $M-1$ for each u. Let $\boldsymbol{Z}$ be the $d \times m$ matrix with $(1, M, M^2, \ldots, M^{m-1})$ and $(1, M+1, (M+1)^2, \ldots, (M+1)^{m-1})$ as two columns, and the rest arbitrary. Conditioning on the sufficient statistic associated with either of these two columns in effect conditions on all of the $\boldsymbol{Y}$, since the $\boldsymbol{Y}$ can be reconstructed from each of these sufficient statistics by themselves. Gibbs sampling in this situation will fail.

Consider a second logistic regression example, in which the first and last components of the sufficient statistic are conditioned on:

Index					N	y	v
1	1	0	0	0	1	1	0
2	1	1	0	0	0	0	0
3	1	0	1	0	0	0	0
4	1	1	1	0	1	0	1
5	1	0	0	1	1	0	0
6	1	1	0	1	0	0	0
7	1	0	1	1	1	0	0
8	1	1	1	1	1	0	0
$\boldsymbol{s} = \boldsymbol{Zy}$	1	0	0	0			
$\boldsymbol{t} = \boldsymbol{Zv}$	1	1	1	0			

No series of rearrangements of the indicators in $\boldsymbol{v}$, each keeping the first, last, and second or third components of $\boldsymbol{s}$ fixed, will draw $\boldsymbol{s}$ closer to $\boldsymbol{t}$.

One might be tempted to try to extend the above argument to discrete distributions on $\boldsymbol{T}$ that will hold asymptotically as the density of discrete points of $\mathfrak{T}$ increases. However, such applications usually involve positive probability on the boundary points of the sets $\mathfrak{Y}_j$, which may leave vertices of $\mathfrak{T}$ forming subsets of the state space not communicating with other state space points.

Instead, combinatoric arguments examining rearrangements of the counts in $\boldsymbol{y}$ are necessary to prove the following theorem:

Theorem 6.2 : For the statistics $\boldsymbol{T}$ of (6), where

1. each $\mathfrak{Y}_j$ is the intersection of a connected subset of $\mathbb{R}$ and $\mathbb{Z}$, each with at least two elements.
2. The matrix $\boldsymbol{Z}$ has a column of ones as its first column, and consists entirely of zeros and ones.
3. There exists a path through the corresponding rows of $\boldsymbol{Z}$, where two rows are connected if they are identical except for one entry.
4. None of the first d components of the sufficient statistic are at their extreme values.

Then the associated Gibbs sampling Markov chain associated with conditioning on the first d entries of $\boldsymbol{T}$ is irreducible.

Corollary 6.3 : The result of Theorem 6.2 holds if condition 3 is replaced by

3. For each row $\boldsymbol{z}$ with a non-fixed unit entry, say in column u, there exists a row $\boldsymbol{w}$ identical to $\boldsymbol{z}$ in $\boldsymbol{Z}$, except that $\boldsymbol{w}$ has a zero in column u, and these pairs exhaust $\boldsymbol{Z}$.

7. Convergence of the Markov Chain

Kolassa (1994b) demonstrates convergence of the Markov chain constructed by Kolassa and Tanner (1994) by showing that certain sets are small in the sense of Definition 2.3, and by using convergence criteria given by Nummelin (1984) to demonstrate the existence of an equilibrium distribution. To apply this definition a dominating distribution must be found. This dominating distribution is expressed in terms of $\hat{\boldsymbol{W}}$, by embedding the Markov chain $\boldsymbol{t}^{(n)}$ in a larger sample space, allowing the quantities $\hat{\boldsymbol{W}}$ to be calculated from sample points, to obtain:

Theorem 7.1 : If $(d\hat{z}/d\hat{w}_1)\hat{z}^{-1}\hat{w}_1^{-1} - \hat{z}\hat{w}_1^{-3}$ is uniformly bounded above by a constant ϵ_1 less than unity, and if it is uniformly bounded below, then the Markov chain $\boldsymbol{T}^{(n)}$ is geometrically ergodic.

Schervish and Carlin (1992) discuss strategies for demonstrating that a Markov chain is geometrically ergodic, and Tierney (1991) discusses strategies for demonstrating both kinds of ergodicity. The following counterexample shows that in a very simple example of Gibbs sampling, the resulting chain is not uniform ergodic, implying that Theorem 7.1 is as strong as one can expect. Tierney (1991) notes that uniform ergodicity holds if and only if the entire sample space is small, and I show that this is not the case.

Suppose $\boldsymbol{T}$ is bivariate normal, parameterized as $N(\mathbf{o}, \boldsymbol{\Sigma})$, and hence $K(\beta) = \frac{1}{2}\beta^T \boldsymbol{\Sigma}\beta$. Express $\boldsymbol{\Sigma}$ as $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$. Applying the standard Gibbs sampler to $\boldsymbol{T}$, the distribution of $\boldsymbol{T}^{(n)}$ conditional on $\boldsymbol{T}^{(n-1)}$ is multivariate normal, $E[\boldsymbol{T}^{(n)}|\boldsymbol{T}^{(n-1)}] = \rho(1,\rho)^T t_2^{(n-1)}$, and $\text{Var}[\boldsymbol{T}^{(n)}|\boldsymbol{T}^{(n-1)}]$ is a positive definite matrix not dependent on $\boldsymbol{T}^{(n-1)}$. Hence for any bounded measurable set $A \subset \mathbb{R}^2$, $P[\boldsymbol{T}^{(n)} \in A|\boldsymbol{T}^{(n-1)} = \boldsymbol{t}^{(n-1)}] \to 0$ as $t_2^{(n-1)} \to \infty$, and hence the entire sample space $\mathbb{R}^2$ is not small.

8. Accuracy of Equilibrium Distributions

Schervish and Carlin (1992) define a space $\mathcal{H} = L^2(\mu/p)$ consisting of the functions f on $\mathfrak{T}$, measurable with respect to $\mathcal{T}$, such that $||f||_{\mathcal{H}} = \int f^2(y)\mu(dy)/p(y) < \infty$, where p is the density of the equilibrium distribution. Then $\mathcal{H}$ is a Hilbert space (Rudin, 1976), and $\mathcal{H} \ni p$. For a transition density P formed from Gibbs sampling, let S be the operator $g \mapsto \int_{\mathfrak{T}} P(t,y)g(t)\mu(dt)$. This operator maps an unconditional distribution on the state space of the Markov chain to the distribution after one iteration. For every $g \in L^2(\mu/p)$,

$$\int g(y)\, dy = \int (Sg)(y)\, dy, \int g(y)\, dy = \int (S^* g)(y)\, dy, \tag{7}$$

where S^* is the adjoint operator to S. These facts are used to show that when S is restricted to the set of functions in $\mathcal{H}$ integrating to unity, the norm of the resulting operator is strictly less than one. Liu, Wong, and Kong (1994) perform similar calculations for the space $L^2(\mu p)$. The Banach space fixed–point theorem can then be used to show that the chain converges geometrically.

In the absence of knowledge that the transition probabilities of the approximate Gibbs sampler correspond to conditional distributions from some unknown joint distribution, the conditional distribution p used to define $\mathcal{H}$ is undefined, and the above construction fails. If, alternatively, another appropriate norming distribution is substituted, the second equality in (7) fails. Hence techniques of §7 were necessary to prove the existence of the equilibrium density p_m.

Once existence of an equilibrium density is demonstrated, Hilbert space techniques might be used to show that the equilibrium distribution p_m approximates p to the proper order. This work is currently in progress.

9. Further Work

Kolassa (1992a) presents the following higher–order double saddlepoint approximation to conditional tail probabilities:

Theorem 9.1 : The second-order saddlepoint approximation to the conditional cumulative distribution function to $O(m^{-5/2})$ is,

$$1 - \tilde{F}(x^1|x^2,\ldots,x^d) = 1 - \Phi(\sqrt{m}\hat{w}_1) + \phi(\sqrt{m}\hat{w}_1)\Bigg(\frac{1}{(\sqrt{m}\hat{w}_1)^3} - \frac{1}{\sqrt{m}\hat{w}_1} + \frac{1}{\sqrt{m}\hat{z}} \times \Big(1 + \frac{1}{m}\big[\tfrac{1}{8}(\hat{\rho}_4 - \tilde{\rho}_4) - \tfrac{1}{8}(\hat{\rho}_{13} - \tilde{\rho}_{13}) - \tfrac{1}{12}(\hat{\rho}_{23} - \tilde{\rho}_{23}) - \tfrac{1}{2}\frac{\hat{\kappa}_{1k}\hat{\kappa}^{ijk}\hat{\kappa}_{ij}}{\hat{\beta}_1} - \frac{\hat{\kappa}^{11}}{(\hat{\beta}_1)^2}\big]\Big)\Bigg),$$

where $\hat{z}$ is given by (2). The invariants are given by $\hat{\rho}_{13}^2 = \hat{\kappa}^{gij}\hat{\kappa}^{hkl}\hat{\kappa}_{gh}\hat{\kappa}_{ij}\hat{\kappa}_{kl}$, $\hat{\rho}_{23}^2 = \hat{\kappa}^{gij}\hat{\kappa}^{hkl}\hat{\kappa}_{gh}\hat{\kappa}_{il}\hat{\kappa}_{jl}$, and $\hat{\rho}_4 = \hat{\kappa}^{ijkl}\hat{\kappa}_{ij}\hat{\kappa}_{kl}$; $\tilde{\rho}_4$, $\tilde{\rho}_{13}^2$, and $\tilde{\rho}_{23}^2$ are the corresponding quantities calculated from $\tilde{\kappa}_{ij}$, $\tilde{\kappa}^{ijk}$, and $\tilde{\kappa}^{ijkl}$.

Kolassa (1992b) considers application of these methods in logistic regression problems when a solution to (3) does not exist. Consider a model for counts of binary outcomes Y_i: For each $i \in \{1,\ldots,M\}$ let Y_i be the number of successes in N_i Bernoulli trials, each with success probability $\pi_i = \exp(\eta_i)/(1+\exp(\eta_i))$, where $\eta_i = \boldsymbol{z}_i\boldsymbol{\beta}$. The quantities $\boldsymbol{z}_i \in \mathbb{R}^d$ are row vectors of covariates. Let $\boldsymbol{Y}$ and $\boldsymbol{N}$ be the vectors of the number of successes Y_i, and the number of binary trials N_i, each with M components. Sufficient statistics $\boldsymbol{T}$ are given by (6). For some $\boldsymbol{Y}$ one or more components of the saddlepoint $\hat{\boldsymbol{\beta}}$ may be infinite. This is likely to happen when the binary outcomes associated with a certain covariate vector $\boldsymbol{z}_i$ are all successes or all failures.

Albert and Anderson (1984) provide a diagnostic for whether all saddlepoint components are finite. Clarkson and Jennrich (1991) determine which covariate vectors are associated with fitted probabilities of 0 or 1. Kolassa (1992b) extends (1) to this case:

Theorem 9.2 : Consider the logistic regression problem above, with sufficient statistics given by (6). Let $\boldsymbol{W}$ be $\boldsymbol{Z}$ with the column corresponding to covariate j removed. Choose $\boldsymbol{s}$ and $\boldsymbol{r}$ in $[0,1]^M$. Maximize the sum of all components of $\boldsymbol{s}$ and $\boldsymbol{r}$ subject to

$$(\boldsymbol{t}(\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T - \boldsymbol{N})\boldsymbol{s} - \boldsymbol{t}(\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T\boldsymbol{r} = \mathbf{o},$$
$$(\boldsymbol{I} - \boldsymbol{W}(\boldsymbol{W}^T\boldsymbol{W})^{-1}\boldsymbol{W}^T)(\boldsymbol{s} - \boldsymbol{r}) = \mathbf{o}.$$

Let $\boldsymbol{s}^*$ and $\boldsymbol{r}^*$ be the maximizing values of these vectors. Classify all observation indices j into one of three mutually exclusive and exhaustive sets as follows: An index j is assigned to $\mathcal{D}_A$ if the maximum above occurs when $r_j^* = 1$. An index j is assigned to $\mathcal{D}_B$ if the maximum above occurs when $s_j^* = 1$. The remaining indices are assigned to to $\mathcal{D}_C$ if the maximum above occurs when $s_j^* = r_j^* = 0$. Let $\boldsymbol{W}_1$ be the matrix of rows of $\boldsymbol{W}$ whose indices are in $\mathcal{D}_A \cup \mathcal{D}_B$, multiplied by -1 if for rows corresponding to indices in $\mathcal{D}_A$, and let $\boldsymbol{W}_2$ be the matrix of the remaining rows of $\boldsymbol{W}$ whose indices are in $\mathcal{D}_C$. Let $\boldsymbol{V}$ be the matrix formed by inserting as row j of $\boldsymbol{W}_2^T$ a vector of zeros. Prepend as the first column of $\boldsymbol{V}$ a vector of zeros except with 1 in coordinate j. Let $\boldsymbol{U}^{\perp T}$ be the result of performing Gram-Schmidt orthonormalization on the columns of $\boldsymbol{V}$. Let $\boldsymbol{U}$ be any matrix with $d - rank(\boldsymbol{V})$ orthogonal columns such that $\boldsymbol{U}^{\perp}\boldsymbol{U}^T = \mathbf{o}$; $\boldsymbol{U}$ may be constructed by completing the Gram-Schmidt process. Solve $\boldsymbol{U}^{\perp T}K'(\hat{\boldsymbol{\beta}};\boldsymbol{t}) = \mathbf{o}$ subject to $\boldsymbol{U}\hat{\boldsymbol{\beta}} = \mathbf{o}$, using Newton iterations of the form

$$\boldsymbol{\beta} - \boldsymbol{\beta}_0 = -(K''(\boldsymbol{\beta}_0;\boldsymbol{t})\boldsymbol{U}^{\perp}\boldsymbol{U}^{\perp T}K''(\boldsymbol{\beta}_0;\boldsymbol{t})$$
$$+ \boldsymbol{U}\boldsymbol{U}^T)^{-1}K''(\boldsymbol{\beta}_0;\boldsymbol{t})\boldsymbol{U}^{\perp}\boldsymbol{U}^{\perp T}K'(\boldsymbol{\beta}_0;\boldsymbol{t}).$$

The vector $\tilde{\boldsymbol{\beta}}$ is obtained similarly. Let $\Sigma_{\boldsymbol{T}}^*(\boldsymbol{\beta}) = \sum_j \boldsymbol{z}_j^T\pi(\boldsymbol{\beta})(1-\pi(\boldsymbol{\beta}))\boldsymbol{z}_i + \boldsymbol{U}^T\boldsymbol{U}$, where the summation is over $j \in \mathcal{D}_C$. Then

$$P(T_j \leq t_j | \boldsymbol{T}_{-j} = \boldsymbol{t}_{-j}) = \Phi(\sqrt{m}\hat{w}) - \phi(\sqrt{m}\hat{w}) \times$$
$$\left(\frac{\sqrt{\left|\Sigma_{\boldsymbol{T}}^*(\tilde{\boldsymbol{\beta}})\right| \left[\Sigma_{\boldsymbol{T}}^{*-1}(\tilde{\boldsymbol{\beta}})\right]_{jj}}}{\hat{\beta}_j\sqrt{m\left|\Sigma_{\boldsymbol{T}}^*(\hat{\boldsymbol{\beta}})\right|}} - \frac{1}{\sqrt{m}\hat{w}} \right) + O(m^{-3/2}).$$

Acknowledgements

The author gratefully acknowledges suggestions from Martin Tanner, Fengchung Peng, the organizers and participants of the Interface session at which this work was presented, and the organizers and participants at the AMS–IMS–SIAM conference on Markov Chain Monte Carlo Methods.

References

Albert, A., and Anderson, J.A. (1984), "Maximum Likelihood Estimates in Logistic Regression," *Biometrika,* 71, 1-10.

Barndorff-Nielsen, O.E., and Cox, D.R. (1979), "Edgeworth and Saddlepoint Approximations with Statistical Applications," *Journal of the Royal Statistical Society, Ser. B,* 41, 279-312.

Besag, J., and Clifford, P. (1989), "Generalized Monte Carlo Significance Tests," *Biometrika,* 76, 633–642.

Besag, J., and Clifford, P. (1991), "Sequential Monte Carlo *p*-values," *Biometrika,* 78, 301–304.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice,* Cambridge, MA: MIT Press.

Clarkson, D.B., and Jennrich, R.I. (1991), "Computing Extended Maximum Likelihood Estimates for Linear Parameter Models," *Journal of the Royal Statistical Society, Ser. B,* 53, 417-426.

Geman, S., and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 6, 721–741.

Kolassa, J.E. (1992a), "Issues in Higher-Order Approximations to Conditional Distribution Functions," *Technical Report 92-05, Department of Biostatistics, University of Rochester.*

Kolassa, J.E. (1992b), "Infinite Parameter Estimates in Logistic Regression," *Technical Report 92-03, Department of Biostatistics, University of Rochester.*

Kolassa, J.E. (1994a), *Series Approximation Methods in Statistics,* New York: Springer – Verlag.

Kolassa, J.E. (1994b), "Convergence of the Gibbs–Skovgaard Algorithm," *Technical Report 94-04, Department of Biostatistics, University of Rochester.*

Kolassa, J.E., and Tanner, M.A. (1994), "Approximate Conditional Inference in Exponential Families Via the Gibbs Sampler," *Journal of the American Statistical Association,* 89, 697–702.

Liu, J., Wong, W.H., and Kong, A. (1994), "Correlation Structure and Convergence Rate of the Gibbs Sampler," *Biometrika,* 81, 27–40.

Lugannani, R., and Rice, S. (1980), "Saddle Point Approximation for the Distribution of the Sum of Independent Random Variables," *Advances in Applied Probability,* 12, 475-490.

Muller, T.P., and Mayhall, J.T. (1971), "Analysis of Contingency Table Data on Torus Mandibularis Using a Log–Linear Model," *American Journal of Physical Anthropology,* 34, 149–154.

Nummelin, E. (1984), *General irreducible Markov chains and non-negative operators,* New York: Cambridge University Press.

Roberts, G.O., and Smith, A.F.M. (1994), "Simple Conditions for the Convergence of the Gibbs Sampler and Metropolis–Hastings Algorithms," *Stochastic Processes and their Applications,* 49, 207-216.

Roberts, G.O., and Polson, N.G. (1994), "On the Geometric Convergence of the Gibbs Sampler," *Journal of the Royal Statistical Society, Ser. B,* 56, 377–384.

Rudin, W. (1976), *Principals of Mathematical Analysis,* New York: McGraw-Hill.

Schervish, M.J., and Carlin, B.P. (1992), "On the Convergence of Successive Substitution Sampling," *Journal of Computational and Graphical Statistics,* 1, 111-127.

Skates, S.J. (1993), "On Secant Approximations to Cumulative Distribution Functions," *Biometrika,* 80, 223-235.

Skovgaard, I.M. (1987), "Saddlepoint Expansions for Conditional Distributions," *Journal of Applied Probability,* 24, 875-887.

Tanner, M.A. (1993), *Tools for Statistical Inference,* Heidelberg: Springer–Verlag.

Tanner, M.A., and Wong, W.H. (1987), "The Calculation of Posterior Distributions by Data Augmentation (with discussion)," *Journal of the American Statistical Association,* 82, 528–550.

Tierney, L. (1991), "Markov Chains for Exploring Posterior Distributions," *Technical Report No. 560, School of Statistics, University of Minnesota.*

ONE-STEP SADDLEPOINT APPROXIMATIONS AND APPLICATIONS IN CONDITIONAL INFERENCE

Suojin Wang*
Department of Statistics
Texas A&M University
College Station, Texas 77843–3143

Abstract

In this paper we propose two simple one-step methods for approximating distribution quantiles of statistics. The methods are derived by approximately inverting saddlepoint formulas for the cumulative distribution functions of the sample mean. The resulting formulas are suitable for use with a pocket calculator. Their extensions to conditional inference and finite population problems are also discussed.

key words and phrases: Conditional distribution, Cornish-Fisher approximation, Finite population, Quantiles, Saddlepoint methods.

1. INTRODUCTION

The problem of approximating the distribution of a statistic is an important one in statistical theory and in practice. The normal, Edgeworth and saddlepoint approximations are the three most commonly used methods. The normal approximation is very simple, but often inaccurate, especially for small sample sizes. The Edgeworth expansions are slightly more complicated, but theoretically more appealing. While they usually improve over the normal approximations, their numerical accuracy is still often questionable. Even worse, they have some undesirable properties, such as negative tail probabilities.

Saddlepoint methods, on the other hand, generally provide accurate approximations whenever they are applicable. They have played an increasingly important role in statistics since its introduction into statistics by Daniels' (1954) pioneering paper, especially during the last decade. See Daniels (1987), Reid (1988) and Field & Ronchetti (1990) for general reviews of the background and development of saddlepoint methods.

In many statistical problems we do not directly use the distribution of a certain statistic. Instead, we need the quantiles of the distribution, which can be obtained by inverting the distribution function. The inversion of the normal approximation is easily accomplished, even on a pocket calculator. Formulas for the inverse of the Edgeworth approximations are available. They are often called Cornish-Fisher approximations; see Witters (1984) for a recent development in this area.

One major disadvantage of saddlepoint approximations has been that their analytic inversions are not available and no quick and easy numerical procedures are developed for practitioners, possibly armed only with a calculator, although lengthier procedures do exist, e.g., the method of bisection search. This, together with the fact that the saddlepoint expansions are based on slightly more mathematical analysis, makes the accurate approximation methods less appealing to practitioners.

Attempting to remedy this drawback, in this note we present two simple one-step methods to compute approximate saddlepoint expansions for quantiles. Using the results of the interesting work of Jensen (1992), two Newton-Raphson type numerical procedures are derived in Section 2 that are easily implemented on a pocket calculator. We show that our one-step methods are generally accurate enough for most applications. Iterative adjustments with quick convergence to the true saddlepoint quantiles are suggested. Section 3 describes an extension of the one-step methods for quantiles of a conditional distribution. Two examples are considered in Section 4 to demonstrate their numerical performance.

Note that the two proposed methods in this paper are aimed to make the saddlepoint approximations more convenient and easier to use, and thus more attractive to practitioners. When there are computers conveniently available, it is more natural to use in a program the well-known methods of bisection and Newton-Raphson to calculate quantiles, although the two new methods developed here are still serious competitors. For example, the second method given in Section 2.2 does not even need to solve (2) for the saddlepoint. This is different from all the existing approaches. Hesterberg (1994) gives

*This material is based in part upon work supported by the Texas Advanced Research Program under Grant No. 160802 and the National Science Foundation under Grant DMS-9200610. A part of the work was performed while the author was visiting the U.S. Bureau of Labor Statistics.

alternative methods for obtaining saddlepoint quantile, distribution function, and inverse distribution function estimates.

2. TWO ONE-STEP SADDLEPOINT APPROXIMATIONS FOR QUANTILES

In this section we consider two one-step saddlepoint methods in the following subsections.

2.1. The first method

Assume that $X_1, \ldots, X_n$ are independent and identically distributed with the moment generating function existing in a non-trivial open interval containing the origin. Further assume that we are interested in approximating quantiles of the distribution of statistic T. In particular, we consider here $T = \overline{X}$, the sample mean. More general cases will be discussed in later sections.

Let $\mu = E(X_1)$ and $K(t) = \ln M(t)$ be the cumulant generating function of $X_1 - \mu$. The Lugannani & Rice (1980) saddlepoint formula for approximating $F_n(x) = \Pr(\overline{X} - \mu \leq x)$ is defined as

$$G_n(x) = \Phi(w_x) + \phi(w_x)(\frac{1}{w_x} - \frac{1}{z_x}), \qquad (1)$$

where

$$w_x = \text{sgn}(t_x)[2n\{t_x x - K(t_x)\}]^{\frac{1}{2}},$$
$$z_x = t_x\{nK''(t_x)\}^{\frac{1}{2}},$$

t_x is the solution to

$$K'(t) = x, \qquad (2)$$

and Φ and ϕ are the standard normal cumulative distribution and density functions, respectively. The relative error in $G_n(x)$ is of order $O(n^{-1})$ in a large deviation region and $O(n^{-3/2})$ in the shrinking set of $\{x : |x - \mu| < c/\sqrt{n}\}$ for any fixed $c > 0$.

Given $\alpha \in (0, 1)$ let x_α be the αth quantile of F_n, i.e., $F_n(x_\alpha) = \alpha$. It is readily shown that the corresponding saddlepoint quantile $x_{s\alpha} = G_n^{-1}(\alpha)$ satisfies

$$x_{s\alpha} = x_\alpha\{1 + O(n^{-3/2})\} \qquad (3)$$

for any fixed α as $n \to \infty$. There are methods available to obtain $x_{s\alpha}$ numerically, but they are all hardly workable on a desk top calculator in general. The goal here is to derive a simple method to approximate x_α that can be calculated on a calculator quickly and easily.

First we state an important result of Jensen (1992) as follows. Let

$$r_x^* = w_x - \frac{1}{w_x}\log(\frac{w_x}{z_x}). \qquad (4)$$

Then we have

$$\Phi(r_x^*) = G_n(x)\{1 + O(n^{-1})\}, \qquad (5)$$

and the error holds uniformly for x in a compact set. Furthermore, for any $c > 0$ the error $O(n^{-1})$ can be replaced by $O(n^{-3/2})$ for $|x - \mu| < c/\sqrt{n}$. This result is essentially Lemma 2.1 of Jensen (1992) and an elegant proof of this is given there.

The basic idea of this paper is to use the transformation (4) to obtain a much simpler relationship between α and an approximate αth quantile by (5). A one-step procedure based on this relationship is described as follows.

Let $x_{J\alpha}$ be the x value such that $r_{x_{J\alpha}}^* = \Phi^{-1}(\alpha)$. It is seen from (5) and (3) that

$$\begin{aligned} x_{J\alpha} &= x_{s\alpha}\{1 + O(n^{-3/2})\} \\ &= x_\alpha\{1 + O(n^{-3/2})\}. \end{aligned}$$

With a reasonable initial value x_0 (see (9), for example), we want to get a one-step approximation x_1 for $x_{J\alpha}$, and therefore for x_α. Let

$$x_1 = x_0 + \Delta x_0, \qquad (6)$$

where Δx_0 is a small adjustment given in (8). By (4) for $x = x_{J\alpha}$

$$\begin{aligned} (r_x^*)^2 &= w_x^2 - 2\ln(\frac{w_x}{z_x}) + \{\frac{1}{w_x}\ln(\frac{w_x}{z_x})\}^2 \\ &= d, \end{aligned}$$

where $d = \{\Phi^{-1}(\alpha)\}^2$. Notice that for fixed α the last two terms of $(r_x^*)^2$ are of order at most $O(n^{-1/2})$. Therefore, for $x - \mu = O(n^{-1/2})$ and thus $t_x = O(n^{-1/2})$ we have

$$\frac{d}{dx}(r_x^*)^2 = \frac{d}{dx}w_x^2 + O(1) = 2nt_x + O(1).$$

Hence, the expansion of $(r_{x_1}^*)^2$ at $x = x_0$ is

$$(r_{x_0}^*)^2 + 2nt_{x_0}\Delta x_0 + O(\Delta x_0), \qquad (7)$$

where t_{x_0} is the solution to (2) for $x = x_0$. Setting (7) to be d we obtain

$$\Delta x_0 = \{d - (r_{x_0}^*)^2\}/(2nt_{x_0}). \qquad (8)$$

Let $q_\alpha = \Phi^{-1}(\alpha)\sigma/\sqrt{n}$ and t_{q_α} be the corresponding solution to (2) for $x = q_\alpha$ (when q_α is not in the domain of x, a suitable substitute is needed), where $\sigma^2 = \text{Var}(X_1)$. We suggest to use the following initial value

$$x_0 = q_\alpha + \{d - (r_{q_\alpha}^*)^2\}/(2nt_{q_\alpha}). \qquad (9)$$

When the solution to (2) can be solved analytically in terms of x our one-step saddlepoint approximation x_1 can be calculated easily on a calculator. Note that in constructing confidence intervals or testing hypotheses we are mostly interested in the quantiles in the two tails rather than around the center of the distribution, so that $\sqrt{n}t_{x_\alpha}$ is bounded away from zero. Note that the first and second terms in (9) are of order $O(n^{-1/2})$ and $O(n^{-1})$, respectively. With this initial value we have that Δx_0 in (8) is of order $O(n^{-3/2})$ since

$$\begin{aligned} d-(r^*_{x_0})^2 &= d-\{(r^*_{q_\alpha})^2+2nt_{q_\alpha}\frac{d-(r^*_{q_\alpha})^2}{2nt_{q_\alpha}} \\ &\quad +O(n^{-1})\}=O(n^{-1}) \end{aligned}$$

Moreover, by expanding $(r^*_{x_1})^2$ at $x=x_0$ (first equation below) and at $x=x_{J\alpha}$ (second equation below) respectively as above, we obtain that

$$\begin{aligned} d+O(\Delta x_0) &= (r^*_{x_1})^2=(r^*_{x_{J\alpha}})^2 \\ &\quad +2nt_{x_{J\alpha}}(x_1-x_{J\alpha})+O(x_1-x_{J\alpha}). \end{aligned}$$

From the first equation we have that $d-(r^*_{x_1})^2 = O(n^{-3/2})$. Since $(r^*_{x_{J\alpha}})^2=d$, it is seen that $2nt_{x_{J\alpha}}(x_1-x_{J\alpha})=O(n^{-3/2})$ and thus

$$x_1=x_{J\alpha}\{1+O(n^{-3/2})\}=x_\alpha\{1+O(n^{-3/2})\}. \qquad (10)$$

Our experience shows that the one-step approximation is numerically accurate enough for most applications. Its numerical accuracy is illustrated in Section 4.

In the case where more accuracy is desirable, we could repeat the one-step method by using the previous approximation x_{i-1} as the initial value of the current step:

$$x_i=x_{i-1}+\Delta x_{i-1}, \quad i=2,3,\ldots, \qquad (11)$$

where

$$\Delta x_{i-1}=\{d-(r^*_{x_{i-1}})^2\}/(2nt_{x_{i-1}}).$$

The series $\{x_i\}$ usually converges quickly to $x_{J\alpha}$ as i increases. When $i=2$, it is seen that $(r^*_{x_2})^2=(r^*_{x_1})^2+2nt_{x_1}\Delta x_1+O(\Delta x_1)=d+O(n^{-2})$. Using the same expansion and by induction we have

$$d-(r^*_{x_{i-1}})^2=O(n^{-(i+1)/2})$$

for $i=2,3,\ldots$. Thus, the argument leading to (10) may be used again so that

$$\begin{aligned} d+O(\Delta x_{i-1}) &= (r^*_{x_i})^2=(r^*_{x_{J\alpha}})^2 \\ &\quad +2nt_{x_{J\alpha}}(x_i-x_{J\alpha})+O(x_i-x_{J\alpha}), \end{aligned}$$

which implies that

$$x_i=x_{J\alpha}\{1+O(n^{-i/2-1})\}. \qquad (12)$$

The fast convergence is evident in the examples in Section 4.

2.2. The second method

The solution to (2) is often not in a simple analytic form and can not be easily solved numerically. In such cases, one-step formula (6) may not be very convenient and thus we suggest a slightly different version as follows.

Let $t_{J\alpha}=t_{x_{J\alpha}}$, the solution to (2) when $x=x_{J\alpha}$. Instead of approximating $x_{J\alpha}$ directly as in (6) and (11) we could approximate $t_{J\alpha}$ first and then get the corresponding approximation for $x_{J\alpha}$. Since r^*_x in (4) can also be viewed as a function of t, we rewrite it as $r^*(t)$. For $x=q_\alpha$ we now define $t_{q_\alpha}=q_\alpha/\sigma^2$ (when it is not in the domain of t, use a reasonable substitute). The following initial value for $t_{J\alpha}$ is suggested

$$t_0=t_{q_\alpha}+[d-\{r^*(t_{q_\alpha})\}^2]/(2nq_\alpha).$$

A small adjustment Δt_0 corresponding to Δx_0 in (8) is obtained as

$$\Delta t_0=[d-\{r^*(t_0)\}^2]/\{2nt_0K''(t_0)\}.$$

Thus the one-step approximation for $t_{J\alpha}$ is given by

$$t_1=t_0+\Delta t_0, \qquad (13)$$

and the corresponding one-step approximation for $x_{J\alpha}$ (and thus for x_α) is $K'(t_1)$. Computational savings may be compounded to use this method in complicated cases where more computational efforts are involved to calculate $K(t)$ and its derivatives.

Similarly, further steps $\{t_i\}$ for $t_{J\alpha}$ and $\{K'(t_i)\}$ for x_α can be obtained by using (13) with initial value t_{i-1}. It can be shown that $K'(t_i)$ and x_i have the same order of the relative error for $x_{J\alpha}$, i.e.,

$$K'(t_i)=x_{J\alpha}\{1+O(n^{-i/2-1})\}$$

for $i=1,2,\ldots$. The proof is similar to that leading to (10) and (12), and is thus omitted here. The second example in Section 4 demonstrates the use of this second approach, among other things.

3. SADDLEPOINT QUANTILES OF CONDITIONAL DISTRIBUTIONS

The simple idea considered in Section 2 can be extended to other saddlepoint approximations. To be more specific, however, in this section we concentrate on the case of approximating conditional distributions explored by Skovgaard (1987). The results of that paper are of special importance and are widely used in various problems, but as in other saddlepoint methods there has been no quick way to obtain the corresponding quantiles.

Let $(X_1,Y_1),\dots,(X_n,Y_n)$ be independent and identically distributed continuous random p-dimensional vectors with the cumulant generating function $K(u,v)$ existing in a neighborhood of the origin, where Y_i and v are $(p-1)$-dimensional. Let

$$\dot{K}_a(u,v)=\frac{\partial}{\partial a}K(u,v),$$

$$\ddot{K}_{ab}(u,v)=\frac{\partial^2}{\partial a\partial b}K(u,v)$$

for a,b being either u or v. Denote the sample mean of (X_i,Y_i) by $(\overline{X},\overline{Y})$. Skovgaard (1987) derived a saddlepoint expansion for the conditional distribution $P(\overline{X}-\mu_y\le x|\overline{Y}=y)$ in the same form as in (1),

$$G_n(x|y)=\Phi(w_{x|y})+\phi(w_{x|y})(\frac{1}{w_{x|y}}-\frac{1}{z_{x|y}}),\qquad(14)$$

where

$$\begin{aligned}w_{x|y}&=&&\operatorname{sgn}(u_x)(2n[u_x(\mu_x+x)+v_xy\\&&&-K(u_x,v_x)-\{v_0y-K(0,v_0)\}])^{1/2},\\ z_{x|y}&=&&u_x\{n|\ddot{K}(u_x,v_x)|/|\ddot{K}_{vv}(0,v_0)|\}^{1/2},\end{aligned}$$

v_0 and (u_x,v_x) are the solutions to the saddlepoint equations

$$\dot{K}_v(0,v_0)=y,\qquad v_0\epsilon R$$

and

$$\dot{K}_u(u_x,v_x)=\mu_y+x,\quad \dot{K}_v(u_x,v_x)=y,\\ (u_x,v_x)\epsilon R^p,$$

respectively. Moreover, $\mu_y=\dot{K}_u(0,v_0)$, and $|A|$ denotes the determinant of matrix A.

Letting

$$r^*_{x|y}=w_{x|y}-\frac{1}{w_{x|y}}\log(\frac{w_{x|y}}{z_{x|y}}),\qquad(15)$$

and using Lemma 2.1 of Jensen (1992), one can show that (5) is true for our new $r^*_{x|y}$ and $G_n(x|y)$. To approximate the αth quantile $x_{\alpha|y}$ of $P(\overline{X}-\mu_y\le x|\overline{Y}=y)$ we first define the corresponding saddlepoint quantile as $x_{J\alpha|y}$ such that $\Phi(r^*_{x_{J\alpha|y}})=\alpha$.

We can now construct a one-step approximation $x_{1|y}$ as in Section 2. Let the initial value be

$$x_{0|y}=q_\alpha+\{d-(r^*_{q_\alpha|y})^2\}/(2nu_{q_\alpha}).$$

Since $\frac{\partial}{\partial x}(w_{x|y})^2=2nu_x$, we have the adjustment analogous to (8):

$$\Delta x_{0|y}=\{d-(r^*_{x_{0|y}})^2\}/(2nu_{x_{0|y}}).$$

The one-step approximation

$$x_{1|y}=x_{0|y}+\Delta x_{0|y}\qquad(16)$$

is accurate up to order $O(n^{-3/2})$ as before.

As in Section 2, further steps $x_{i|y}$ $(i=2,3,\dots)$, when needed, are easily defined as well as computed to improve the accuracy until the convergence to the limit $x_{J\alpha|y}$. The same relative error rate of $O(n^{-i/2-1})$ also holds. Furthermore, the second one-step method (13) can be readily extended analogously to the conditional distribution problem. We omit the details.

4. EXAMPLES

In this section we consider two examples to illustrate how our one-step methods are quickly implemented and the numerical accuracy they obtain.

Example 1. Suppose that $X_1,\dots,X_n$ is a sample drawn from the gamma distribution $G(\theta,\beta)$, i.e.,

$$f(x)=\frac{\beta^\theta}{\Gamma(\theta)}x^{\theta-1}e^{-\beta x},\quad x>0,$$

and $T=\frac{1}{n}\sum_{i=1}^n X_i$. It is computationally convenient to make comparisons in this example, since the exact distribution of T is known to be $G(n\theta,n\beta)$. The cumulant generating function of $X_1-\mu$ is

$$K(t)=-\theta\log(1-\frac{t}{\beta})-\frac{\theta}{\beta}t,$$

with the solution to (2) $t_x=\beta-\theta/(x+\theta/\beta)$.

To illustrate the performance of the one-step method defined in (6), (8) and (9) we take $\theta=0.1$, $\beta=1$, a somewhat extreme case. Furthermore, let the sample size n be as small as 10. This choice is also convenient since $\overline{X}$ here has an exponential distribution with mean 0.1, so that the αth quantile of $\overline{X}$ is in fact explicitly given by $x_\alpha=-\log(1-\alpha)/10$. On the other hand, this particular case is representative of the whole gamma family for the comparisons of several methods.

In Table 1, the one-step approximation x_1 is compared with other approximations. Note that when $q_\alpha<0$ we replaced it with 0.01. It is seen that x_1 is close enough to saddlepoint approximation $x_{J\alpha}$ and thus to exact x_α for most practical purposes. In some cases, a few more steps may be desirable and they are easily implemented.

Example 2. This example is to demonstrate that one-step methods are also useful in finite population problems. Let N be the population size and $M_1(t)$ be the moment generating function. A random sample $X_1,\dots,X_n$

α	exact	one-step	2nd step	exact SA	normal	two-term
	x_α	x_1	x_2	$x_{J\alpha}$		Cornish-Fisher
0.005	0.00050	0.00036	0.00046	0.00048	-0.1576	0.0329
0.01	0.00101	0.00098	0.00098	0.00098	-0.1326	0.0247
0.05	0.00513	0.00489	0.00502	0.00504	-0.0645	0.0120
0.1	0.0105	0.0104	0.0104	0.0104	-0.0282	0.0123
0.2	0.0223	0.0212	0.0219	0.0222	0.0158	0.0208
0.8	0.1609	0.1674	0.1589	0.1618	0.1842	0.1597
0.9	0.2303	0.2312	0.2315	0.2315	0.2282	0.2305
0.95	0.2996	0.2995	0.3014	0.3011	0.2645	0.3017
0.99	0.4605	0.4600	0.4630	0.4626	0.3326	0.4694
0.995	0.5298	0.5294	0.5325	0.5322	0.3576	0.5428

Table 1: Approximations to quantiles of $T = \overline{X}$ in *Gamma*(0.1, 1) case; $n = 10$

($n \leq N$) is drawn from the population. We wish to approximate the quantiles of $\overline{X} - \mu$. Notice that the distribution of the $\overline{X}$ is generally significantly different from that of the mean of a sample drawn with replacement. The bootstrap is based on the latter sampling scheme. See Davison & Hinkley (1988) for an interesting account of saddlepoint approaches in this area where our one-step methods are applicable as in the infinite population case.

The one-step methods described in Section 2 may be applied to our current problem with some modifications. Let $K_n(t)$ be the cumulant generating function of $\overline{X} - \mu$. Then $K_n(t)$ can be expressed in terms of $M_1(jt/n)$ ($j = 1, 2, \ldots, n$) and computed recursively. Moreover, if we use $R_n(t) = \frac{1}{n}K_1(nt)$ to replace $K(t)$ in Section 2, then with a negligible error due to discreteness, the saddlepoint approximation is still valid as $n, N \to \infty$ and $n/N \leq d < 1$ for some constant d. These results have been given in Wang (1993).

We now compare the one-step approximation obtained from (13) with the exact quantiles and other approximations. To carry out the comparisons, the following population with $N = 36$ was simulated from an exponential distribution:

4.295	34.636	11.204	3.041	3.694	0.570
31.024	11.245	8.591	12.745	6.568	3.913
18.615	6.332	6.841	0.905	13.610	14.981
10.103	2.210	0.765	5.056	7.038	1.849
1.594	18.450	1.591	6.656	22.752	12.753
0.790	5.005	7.418	11.321	5.631	3.834

Table 2: Population for the simulation in Example 2

It is obtained that $\mu = 8.8230$. Table 3 lists approximations to quantiles of $\overline{X} - \mu$ with $n = 5$. It is seen that the one-step method provides good approximations for the quantiles. Explicit formulas for Cornish-Fisher type approximations in the finite population case are not available. Thus, they are not given in the table.

α	'exact'	one-step	2nd step	exact SA	normal
	x_α	$K'(t_1)$	$K'(t_2)$	$x_{J\alpha}$	
0.005	-6.253	-6.126	-6.283	-6.296	-8.551
0.01	-5.898	-5.810	-5.935	-5.945	-7.722
0.05	-4.766	-4.707	-4.781	-4.788	-5.460
0.1	-4.020	-3.958	-4.027	-4.034	-4.254
0.2	-2.981	-2.879	-2.945	-2.957	-2.794
0.8	2.863	2.908	2.841	2.840	2.794
0.9	4.630	4.735	4.628	4.623	4.254
0.95	6.113	6.242	6.123	6.123	5.460
0.99	8.961	8.996	8.933	8.929	7.722
0.995	9.940	9.976	9.933	9.929	8.551

Table 3: Approximations to quantiles of $T = \overline{X} - \mu$ in Example 2 with $n = 5$. 'Exact' distribution based on 100,000 simulated samples

5. Concluding Remarks

In this note we have proposed two simple one-step saddlepoint methods for distribution and conditional distribution quantiles. We have also discussed their applications in finite population problems which are common in survey sampling and other areas. The one-step approximations are easily computed on a pocket calculator once the cumulant generating function is available. Again, we stress that in presenting the new methods here, their simplicity has been the main objective. It is indeed an important step to make great methods easily usable to attract the interest of practitioners in using them.

The new methods were developed in the cases where sample means are the statistics under consideration. However, since Jensen's (1992) original theoretical results that we have applied here are valid for many other more complicated problems, the one-step methods can be obtained similarly in those cases. Finally we note that our experience reveals that while the second method (13) is often more convenient, for it does not require to solve (2), it is generally slightly less accurate than the first method (6). This is because extra approximations are involved in the second method. Therefore we recommend using the first method whenever the solution to (2) is easily calculable.

6. References

Daniels, H. E. (1954). Saddlepoint approximations in statistics. *Ann. Math. Statist.* **25**, 631–650.

Daniels, H. E. (1987). Tail probability approximations. *Int. Statist. Rev.* **55**, 37–48.

Davison, A. C. & Hinkley, D. V. (1988). Saddlepoint approximations in resampling methods. *Biometrika* **75**, 417–431.

Field, C. A. & Ronchetti, E. (1990). *Small Sample Asymptotics.* Hayward, CA: IMS Monograph series.

Hesterberg, T. C. (1994). Saddlepoint quantiles and distribution curves, with bootstrap applications. *Computational Statistics*, to appear.

Jensen, J. L. (1992). The modified signed likelihood statistic and saddlepoint approximations. *Biometrika*, **79**, 693-703.

Lugannani, R. & Rice, S. O. (1980). Saddlepoint approximation for the sum of independent random variables. *Adv. Appl. Prob.* **12**, 475-490.

Reid, N. (1988). Saddlepoint methods and statistical inference. *Statist. Sci.* **2**, 213–238.

Skovgaard, I. (1987). Saddlepoint expansions for conditional distributions. *J. Appl. Prob.* **24**, 275–287.

Wang, S. (1993). Saddlepoint expansions in finite population problems. *Biometrika*, **80**, 583-590.

Witters, C. S. (1984). Asymptotic expansions for distributions and quantiles with power series cumulants. *J. R. Statist. Soc.* B **46**, 389–396.

ESTIMATION OF COVARIANCE MATRICES USING EIGENSTRUCTURE INFLUENCE

John E. Vetter
Naval Air System Command
Washington D.C.

Robert W. Jemigan
The American University
Washington D.C.

We consider robust and resistant estimation of the covariance matrix of multivariate data. Using expansions for the influence of individual observations on the eigenvalues and eigenvectors of the covariance matrix derived by Critchley(1985), we develop an Eigenstructure Influence (ESI) method for estimating the covariance matrix. Motivated by lower rank approximations and graphical representations of covariance matrices, we extend the influence measures developed by Critchley to measure the influence individual observations may have on these approximations. We develop a downweighted, iterative estimation algorithm estimating the covariance matrix directly using the ESI influence measure. We illustrate the technique with sample data sets and lower rank biplot graphics.

KEY WORDS: Eigenvalues; Eigenvectors; Influence functions

1. INTRODUCTION

We consider a robust and resistant procedure to estimate the covariance matrix of multivariate data based upon the eigenstructure influence functions. The motivation behind the use of influence functions is that they measure the amount of change in parameter estimates at a point $\mathbf{x} \in \mathbb{R}^p$. Outlying points $\mathbf{x}$ do not necessarily unduly influence parameter estimates, while conversely, non-outlying points with large influence may change parameter estimates by nature of their orientation. Distributional properties of the sample influence functions, graphical methods for the identification of influential points, using influence of both the eigenvalues and the eigenvectors, and specific application to biplots and other lower rank applications are discussed by Vetter (1992). The procedure will be referred to as the Eigenstructure influence (ESI) procedure. Its performance is compared with other methods available in the literature.

2. MOTIVATION

Exploratory and robust/resistant techniques are becoming a more widely accepted component of statistical practice. Estimation based upon a sample of points is often enhanced by the use of robust/resistant methods. These methods attempt to ameliorate the problem of distortion of the underlying structure of the main body of observations by a small number of observations. In addressing this problem there have been many creative and innovative ideas presented to handle what is frequently called contamination. Contamination can take many forms but basically can be defined as any point or set of points which unduly influence the outcome of an analysis or investigation. Specifically, in the estimation of dispersion matrices for a set of multivariate vector observations, there have been numerous schemes to ameliorate the effects of outlying points, roundoff errors and other forms of distortion which can be present in any set of observed data. Many of these schemes involve the use of Mahalanobis distance which measures the elliptical distance from a multivariate centroid. Jolliffe (1986) has suggested that the influence functions on both the eigenvectors and the eigenvalues be used instead of the Mahalanobis distance. The ESI procedure provides a systematic method of using both these influence functions, as Jolliffe has suggested.

The classical estimator for dispersion matrices is the maximum likelihood estimator (MLE), which has optimal properties when the data observed are from a multivariate normal distribution with a given mean vector and covariance matrix. The maximum likelihood estimator may not possess these optimal properties, however, when contamination is present in the data.

In any analysis, one must determine whether contamination is present or not. For multivariate data it can be extremely difficult to identify the contaminated portion of data. The problem is particularly acute in small data sets, where the identification of the population distribution may be extremely difficult.

3. MAHALANOBIS DISTANCE

The **Mahalanobis** distance, as presented in the following discussion, has been the basis of most multivariate estimation procedures to date. The Mahalanobis distance quantifies in a scalar measure the elliptical distance from a vector centroid. Once the centroid vector is determined, the distance of a point in p dimensions is measured and is weighted by the inverse of the covariance matrix. Thus, points which lie along an axis with a large amount of variation may have the same elliptical distance as those points which lie closer in Euclidean distance to the centroid but lie along an axis with less variation. This measure has been proven to be an effective measure of outlyingness and has been used very successfully in the iterative procedures to detect outlying points.

The ESI procedure is based upon the influence functions of the eigenvalues and eigenvectors of the covariance matrix which measure the rate of change in the eigenvalues and eigenvectors at a point **x**. It will be shown that these rates of change, though related to Mahalanobis distance provide more information and thus can be used to improve the estimation of covariance in the presence of contamination. The Mahalanobis distance is defined as:

$$M_i = (x_i - \hat{\mu})' \, \hat{\Omega}^{-1} \, (x_i - \hat{\mu})$$

where it is assumed that

$$X_i \sim N(\mu, \Omega)$$

and

$$\hat{\Omega} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})(x_i - \hat{\mu})'$$

To date, techniques have been primarily focused on outlyingness of points and not on the influence of points on particular parameters. Since there are many forms of data contamination, the proposed procedure is offered as an alternative approach which considers all forms of contamination, i.e. any data which distort the parameters of interest.

2. THE ESI METHOD

It has been shown that the Mahalanobis distance is an increasing function of the eigenvalue influence functions of the covariance matrix (see Vetter (1992)). The existing methods, which use Mahalanobis distance are, in effect, using only the influence on the variation (eigenvalue), but not the influence on the eigenvector. To see why the consideration of the influence of the eigenvector as well as the influence on the eigenvalue is important, consider the following decomposition of a covariance matrix **S**:

$$S = \sum_{j=1}^{p} \lambda_j \alpha_j \alpha_j'$$

where α_j are the eigenvectors and λ_j the eigenvalues of **S**. It is logical therefore, to consider the influence of a point on this particular combination of both the eigenvalues and the eigenvectors.

The rank r approximation to **S** is

$$S_{(r)} = \sum_{j=1}^{r} (\sqrt{\lambda_j}\alpha_j)\,(\sqrt{\lambda_j}\alpha_j')$$

Therefore, the influence on the vector functionals

$$\sqrt{\lambda_j}\;\alpha_j \qquad j = 1 \ to \ p$$

can be used to develop a metric which measures the change in the covariance matrix **S** or any lower rank approximation to **S**.

From Critchley (1985), we have the following perturbed parameters where ε_i represents the amount of contamination in the jth eigenvalue and eigenvector at a point x_i :

$$\lambda_j(\epsilon_i) \;=\; \lambda_j+\epsilon_i v_{ij}+\frac{1}{2}\epsilon_i^2\pi_{ij}+0\,(\epsilon_i^3)$$

and

$$\alpha_j(\epsilon_i) \;=\; \alpha_j+\epsilon_i\beta_{ij}+\frac{1}{2}\epsilon_i^2\gamma_{ij}+0\,(\epsilon_i^3)$$

where for each point i, the influence function or relative rate of change in the jth functional $\lambda_j(\varepsilon_i)$ is

$$v_{ij} = y_{ij}^2 - \lambda_j$$

and the influence function for the functional $\alpha_j(\varepsilon_i)$ is

$$\beta_{ij} \;=\; -y_{ij}\sum_{k\neq j}\frac{y_{ik}}{(\lambda_k-\lambda_j)}\alpha_k$$

where

$$y_{ij} = x_i'\alpha_j$$

is the jth principal component score for the ith point.

We have developed the following equation for the perturbed vector $\sqrt{\lambda}(\varepsilon_i)\,\alpha(\varepsilon_i)$

$$\sqrt{\lambda_j(\epsilon_i)}\,\alpha_j(\epsilon_i) = \sqrt{\lambda_j}\alpha_j+\epsilon_i\left(\frac{v_j}{2\sqrt{\lambda_j}}\alpha_j+\sqrt{\lambda_j}\beta_j\right)+0\,(\epsilon_i^2)$$

The empirical influence curve, which measures the rate of change in $\sqrt{\lambda_j}\;\alpha_j$ at the ith point can now be written as the following vector:

$$\hat{I}_{ij}(g) \;=\; \frac{v_{ij}}{2\sqrt{\lambda_j}}\alpha_j \;+\; \sqrt{\lambda_j}\beta_{ij}$$

It should be noted from the form of these influence functions that there are two independent components. The first component

$$\frac{v_{ij}}{2\sqrt{\lambda_j}}\alpha_j$$

will be large when a point is outlying in the jth direction. The second component

$$\sqrt{\lambda_j}\quad\beta_{ij}$$

will be large when a point is influential on the direction of the jth principal component direction. When β is large, the second part may dominate the value of the influence function. Since v_{ij} is independent of β_{ij}, points which are not outlying with respect to elliptical distances may yet be influential for a particular functional.

In order to use these influence functions in a weighted estimation procedure we need to convert the vectors to some scalar measure of influence. A natural scalar metric for measuring change to **S** or any lower rank approximation to **S** is the following:

$$dv_i = \sum_{j=1}^{r} c_j \left\| \frac{v_{ij}}{2\sqrt{\lambda_j}}\alpha_j + \sqrt{\lambda_j}\beta_{ij} \right\|$$

where r is the rank of the estimated matrix and c_j are scaling factors.

A scaling factor which allows a direct comparison with the Mahalanobis distance is:

$$c_j = \frac{1}{\sqrt{\lambda_j}}$$

We then have, a comparison of the functional form of dv to the Mahalanobis distance dm as follows:

$$dm_i = (x_i - \bar{x})'\Omega^{-1}(x_i - \bar{x}) = \sum_{j=1}^{p}\left(\frac{v_{ij}}{\lambda_j} + 1\right)$$

and

$$dv_i = \sum_{j=1}^{p} \left\| \frac{v_{ij}}{2\lambda_j}\alpha_j + \beta_{ij} \right\|$$

If a point has a large Mahalanobis distance it will also have a large dv score, due to the first part which measures influence on the eigenvalue. The dv score also includes the influence on the direction of the eigenvectors, thus protecting against influential but not necessarily outlying points.

4. ESTIMATION PROCEDURE:

Any M-type estimation procedure may be used with weights based upon the dv metric. For example, an accepted and frequently used method is to apply a Huber influence function which gives equal weight to the middle portion of the d values but gives decreasing weights as the scores increase beyond a specific cutoff point k.

Let $w_i = 1$ if $dv_i < k$
$w_i = k/dv_i$ if $dv_i > k$

For an iterative procedure, it is necessary to extend Critchley's equations to determine the influence of a point on a weighted estimate of the covariance matrix. Critchley's equations reflect the influence of a point as the difference between the covariance matrix with the point included and with the point deleted. In our iterative procedure the influence of a point is reflected as the difference between giving a point weight w_i, where $0 < w_i < 1$, and giving it weight 0.

Let

$$m = \sum_{i=1}^{n} w_i x_i$$

denote a weighted mean and

$$\Omega(\hat{F}) = \sum_{i=1}^{n} w_i (x_i - m)(x_i - m)'$$

denote a weighted covariance matrix. Then the weighted average with the ith point downweighted is denoted by

$$m_{(i)} = m - \left(\frac{w_i}{(1-w_i)}\right)(x_i - m)$$

$$= \frac{(m - w_i x_i)}{(1-w_i)}$$

The empirical distribution function after downweighting the ith point becomes

$$\hat{F}_{(i)} = \left\{1 + \frac{w_i}{(1-w_i)}\right\}\hat{F} - \frac{w_i}{(1-w_i)}\delta_i$$

we then get the downweighted covariance matrix

$$\Omega(\hat{F}_{(i)}) = \Omega(\hat{F}) - (\frac{w_i}{1-w_i})\{(x_i-m)(x_i-m)'-\Omega(\hat{F})\}$$

$$-(\frac{w_i}{(1-w_i)})^2(x_i-m)(x_i-m)'$$

Thus, w_i / (1 - w_i) replaces ε_i in the previous equations. We have also shown the equations for the downweighted eigenvalues and eigenvectors and their corresponding influence function. If λ_{wj} and α_{wj} represent the weighted eigenvalue and eigenvector then the formulas for v_{ij} and β_{ij} are the same as deleted influence with y_{ij} = $(x_i$-m)' α_{wj}.

5. RESULTS

It has been shown (see Vetter, 1992) that the ESI procedure consistently outperformed several of the well known procedures based upon the Mahalanobis distance. The bias and mean squared error in estimating covariances and eigenvalues using the ESI procedure was compared to two M-type estimators (see Maronna, 1976) and (Campbell, 1980) both based upon ellipsoidal distances. ESI performed as well as the two procedures when contamination existed in the form of outliers in the direction of the principal component axes, but provided visibly better results when contamination was present between the axes where point have more influence on the eigenvectors.

Another advantage of the ESI procedure is that for problems of less than full rank, the construction of the proposed procedure facilitates improved estimation by exclusion or downweighting of the influence of points on directions not included in the analysis. As an example, in situations where the relationships of the minor principal components are of interest, the influence of points on only these last few components with the smallest variation need be considered. For graphical procedures such as the biplot, a robust/resistant biplot can be obtained by using the ESI method with scaling factors c_1 and c_2 = 1 and c_r = 0 for r > 2. This bases the weights on the influence of points on the 1st 2 principal components upon which the biplot is based, (see Vetter, 1992).

The proposed procedure can be applied to correlation matrices by substituting the formulas for influence of vector observations on the correlation matrix, which have been developed by Calder and described in Jolliffe (1986). This would be particularly beneficial for principal components based upon the correlation matrix, since it has been shown that points which are influential for the covariance matrix need not be particularly influential for the correlation matrix. This is in part due to the fact that for a correlation matrix the eigenvalues sum to the number of variables in the observation vector. An investigation could be made of those points where large influence is indicated for either covariance or correlation but not the other.

REFERENCES

Calder, P., Jolliffe, I. T. and Morgan, B.J.T. (1986). Influential observations in principal component analysis: a case study. Submitted for publication.

Campbell, N.A. (1980). Robust procedures in multivariate analysis, *Applied Statistician* 29, 231-237.

Critchley, F. (1985). Influence in principal components analysis, *Biometrika*, 72, 627-636.

Gabriel K. R. (1971). The biplot-graphic display of matrices with applications to principal components analysis, *Biometrika*, 58, 453-467.

Huber, P. J. (1977). *Robust statistical procedures*, Society for Industrial and Applied Mathematics.

Jolliffe, I. T. (1986). *Principal Components Analysis.* Springer-Verlag, New York.

Maronna, R.A. (1976). Robust M-estimators of multivariate location and scatter. *The Annals of Statistics, 4, 51-67.*

Sibson, R. (1979). Studies in the robustness of multidimensional scaling: perturbational analysis of classical scaling. *Journal of the Royal Statistical Society, Series B*, 41, 217-229.

Vetter, J. E. (1992). Estimation of Covariance Matrices using influence of eigenvalues and eigenvectors. Doctoral dissertation , The American University.

Finding the Minimum Volume Ellipsoid

Wendy L. Poston
Naval Surface Warfare Center, G33, Dahlgren, VA 22448-5000
wposton@pooh.nswc.navy.mil

Carey E. Priebe
Johns Hopkins University, Baltimore, MD 21218
priebe@brutus.mts.jhu.edu

Abstract

Prior to performing an analysis on data, the statistician must address the problem of outliers. One way to do this is to use the Minimum Volume Ellipsoid (MVE) estimator, which has desirable robustness properties due to its breakdown point of 50%. It is defined as the ellipsoid of minimum volume that covers half of the data points. A major problem with using the MVE is that few computationally attractive methods exist for its calculation, especially in high dimensions and for large sample sizes. Determining the MVE consists of two parts. The first is to find the correct subset of points used to calculate the MVE, and the second is to find the ellipse that covers this set. Finding the subset of points to be covered by the MVE will be addressed in this paper. The solution proposed here is to use the Effective Independence Distribution (EID) method which chooses the subset by minimizing determinants of matrices based on the data. Results show that the volume of the ellipse using the EID subset of points differs from the optimal by less than 6% for some regression data sets where the true MVE is known.

1. Introduction

The existence of outliers and how to deal with them is an important problem in statistics. The MVE was first proposed as a robust estimator of location and shape by Rousseeuw [1], but its use has been hampered by the lack of a computationally feasible means of calculating it. The MVE is defined as the ellipsoid of minimum volume that covers approximately half of the points in a data set. From this one can see that it is a configuration of high content, but minimum volume.

The problem of finding the MVE is two-fold. One must first find the subset of points that should be covered by the ellipsoid and then weight the data such that these points are covered by it. A solution for finding the weights is described in Hawkins [2]. Several methods have been described in the literature to find the subset of points. The first was the basic resampling method suggested by Rousseeuw and Leroy [3]. Subsequent methods that have been developed include the Feasible Solution Algorithm (FSA) by Hawkins [2], and some heuristic search algorithms are described in Woodruff and Rocke [4]. These authors compare the resampling or undirected random search method to simulated annealing, genetic algorithms, and tabu search. All of these methods are approximate ones, so obtaining the exact MVE is not guaranteed for a finite amount of resampling.

This paper is based in part on the work done by Hawkins [2]. We propose a solution to the subset selection problem called the EID method. Some background information on the MVE estimator is provided, and the EID method is described. Results are presented that show the relative error in the volume of the ellipsoid found using the EID approach for several regression data sets where the true MVE is known.

2. Minimum Volume Ellipsoid Estimator

The problem of robust estimation of multivariate location and shape is that given a set of n observations $\mathbf{x}_i$, each one having p dimensions, find an estimate of location and shape that is resistant to outliers or contaminated data. The MVE is one such estimator and it is given by the ellipsoid [2]

$$(\mathbf{x}-\mathbf{c})^T\Gamma^{-1}(\mathbf{x}-\mathbf{c}) = p \qquad (1)$$

where $\mathbf{c}$ and Γ are the location vector and scatter matrix respectively and p is the dimension of the data. The location vector is a weighted mean calculated as

$$\mathbf{c} = \sum_{i=1}^{h} w_i \mathbf{x}_i^* \tag{2}$$

and the covariance or scatter matrix is

$$\Gamma = \sum_{i=1}^{h} w_i (\mathbf{x}_i^* - \mathbf{c})(\mathbf{x}_i^* - \mathbf{c})^T \tag{3}$$

where $\mathbf{x}_i^*$ is a column vector denoting the *ith* observation in the subset of *h* points, w_i is the weight for the *ith* observation, and $h = [(n+p+1)/2]$ (the brackets denote the greatest integer function). The volume of the covering ellipse will be proportional to the determinant of Γ. It is evident from Eqs. 2-3 that to find the MVE one must determine which *h* points should be covered and the corresponding weights to ensure coverage of the points.

It is known [1,2,4] that the MVE has a breakdown point that approaches 50% as the number of points in the data set increases which is the best one can have. This means that approximately half of the data can be arbitrarily contaminated without affecting the estimate.

The algorithm used in this paper to find the weights is one developed by Titterington [5] and is also used by Hawkins [2]. All of the weights are initially set to $w_i^{(0)} = 1/h, \ i = 1,\ldots,h$ which is just the usual weights given to points when calculating the sample mean of a data set of size *h*. Then at each iteration *k* calculate the weighted mean and covariance from Eqs. 2-3 and the Mahalanobis distances for each observation given by

$$D_i^{(k)} = (\mathbf{x}_i^* - \mathbf{c}^{(k)})^T \Gamma_{(k)}^{-1} (\mathbf{x}_i^* - \mathbf{c}^{(k)}) \tag{4}$$

If $D_i^{(k)} \leq p$ for every *i*, then the current ellipsoid using $\mathbf{c}^{(k)}$, and $\Gamma_{(k)}^{-1}$ is the MVE covering the *h* observations. If the Mahalanobis distance for any of the observations exceeds *p*, then the weights must be adjusted. They are updated using the following

$$w_i^{(k+1)} = w_i^{(k)} \frac{D_i^{(k)}}{p} \tag{5}$$

and the calculations of Eqs. 2-4 are repeated until all of the distances are less than *p*. This procedure enlarges the ellipsoid until all of the *h* points are covered.

The algorithm for finding the weights can be somewhat computationally intensive for some data sets. However, it should be apparent that the real computational burden arises from the determination of which points should be covered by the ellipse. The EID algorithm is presented as a means of addressing this problem.

3. Effective Independence Distribution

3.1 Background

Since the volume of the minimum covering ellipse is proportional to the determinant of the scatter matrix Γ, one could approach this problem as that of optimizing the determinant. In this application, the objective would be to minimize the determinant of Γ. This provides the motivation for using the EID method, since it can be shown that deleting points based on their EID value will optimize the determinant of the Fisher Information Matrix (FIM) defined below. Of course, the FIM is not exactly the same as Γ of Eq. 3, however results indicate that it will be a reasonable approximation.

The EID vector [6,7,8] for a data set of *n p*-dimensional observations is calculated using the following equation

$$\text{EID} = \mathit{diag}(\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T) \tag{6}$$

where **X** is an *n* x *p* matrix with *n*>>*p* and each row contains one observation. The EID is just the diagonal elements of the 'hat' matrix which is familiar from regression theory. Note that there are *n* elements in the EID vector, one corresponding to each observation. Finally notice that

$$\sum_{i=1}^{n} \text{EID}_i = p \tag{7}$$

and that

$$0 \leq \text{EID}_i \leq 1 \tag{8}$$

which can be shown from the fact that the matrix in Eq. 6 is idempotent. The matrix $\mathbf{X}^T\mathbf{X}$ is called the FIM.

It has been shown [7,8,9] that the following relationship between the determinants of the FIM holds as one observation is deleted from the data set

$$\left|\mathbf{X}^T\mathbf{X}\right|_{-i} = (1 - \text{EID}_i)\left|\mathbf{X}^T\mathbf{X}\right| \tag{9}$$

where the determinant on the left-hand side is calculated with the *ith* observation removed from the data set, the determinant on the right-hand side contains all of the data,

Table I. Data Set Parameters

Data Set	p	n	h
Aircraft	4	23	14
Coleman	5	20	13
Delivery	2	25	14
Education	3	50	27
Gravity	5	20	13
Salinity	3	28	16

and the EID_i denotes the EID value for the *ith* observation. From this one can see that there is a direct relationship between the determinants as the points are removed. Thus, if the situation calls for minimizing the determinant of the FIM then it is obvious from Eq. 9 that the observation with the largest EID value should be deleted.

Two things should be noted from Eqs. 8-9. If an observation is deleted that has a value of zero, then nothing is lost by removing that point. If an observation has an EID value of one, then that point cannot be removed. If such a point is removed, then the determinant of the FIM becomes zero and the resulting matrix is singular. Thus, an observation with a value of one must be retained to keep the problem at full rank p.

3.2 The EID method of subset selection

The EID values can be used to successively remove points from the data set until h points remain. These h points will then be used with the algorithm described in Section 2 that will find the weights and the resulting ellipsoid. However, to better approximate the matrix Γ, the data will be centered by subtracting the p-dimensional sample mean from each observation. This is repeated as each point is deleted. The procedure consists of the following steps:

1. Calculate the matrix

$$\mathbf{X}'^{(j)} = (\mathbf{X}^{(j)} - \overline{\mathbf{X}}^{(j)})$$

where $\mathbf{X}^{(j)}$ is the set of raw data points at the *jth* iteration (at iteration j=0 there are n points in the set, at iteration j=1 there are n-1 points, etc.) and $\overline{\mathbf{X}}^{(j)}$ is an $(n$-$j)$ x p matrix with each row containing the p-dimensional sample mean for the current set of data.

2. Use the matrix $\mathbf{X}'^{(j)}$ in Eq. 6 to calculate the EID value for each point in the current data set.

3. Delete the point that corresponds to the maximum EID value.

4. Repeat steps 1-3 until only h points remain.

5. Adjust the weights until the h points are covered.

The EID tends to give points a large EID value if they have large magnitudes. However, this is not always the case; e.g., if an observation must be retained to keep the problem non-singular then it will have an EID value of one regardless of the magnitude. For a detailed discussion of this point and some examples see Kammer [6] and Poston, Priebe and Holland [8]. For this reason and because the desired output is a robust estimation of location, the centering of the data at each iteration is needed, which is the reason for the first step.

4. Applications and Results

To test the usefulness of this method, it is applied to several data sets where the true MVE is known. The paper by Hawkins [2] gives the correct subset and the resulting volume of the true MVE for these data sets. The relative error in the volume of the ellipse based on the subset obtained using the EID method can then be determined for comparison purposes. There are 6 data sets which are taken from Rousseeuw and Leroy [3]. These data are used for regression purposes, and only the predictors are used here to determine the MVE. The parameters of interest are shown in Table I. From this one can see that the data sizes are relatively small ranging in size from

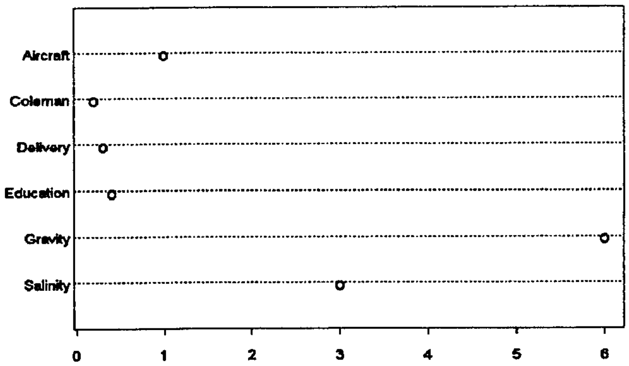


Figure 1. Percent relative error in the volume of the MVE as determined by the EID approach.

Table II. Timing Results for Methods (sec)

Data Set	EID	Splus, Genetic Algorithm
Aircraft	0.22	68.0
Coleman	0.17	67.0
Delivery	0.11	28.0
Education	0.77	74.0
Gravity	0.11	62.0
Salinity	0.22	50.0

n=20 to n=50. The dimensionality of the data is also low, from 2 to 5 dimensions.

For this study, the EID algorithm is implemented in MATLAB on a 486, 33 MHz computer. The relative error in the volumes of the minimum covering ellipsoid using the EID approach is shown in Figure 1. It is evident from the small error that ours is a feasible approach to finding the MVE.

The time needed to determine the subset of points is given in Table II. Also in this table are some timings obtained using Splus 3.1 to determine the MVE estimate of a covariance matrix. This software uses a genetic algorithm to find the subset of points. The purpose here is to provide a very rough comparison of the two methods in terms of the computational effort involved. From these results, one can see that using the EID provides a savings in time when calculating the MVE. This would become more important as the dimensionality and size of the data set increases.

The 2-dimensional 'delivery' data set is shown in Figure 2 to provide a qualitative assessment of the method. From this, one can see that the bulk of the data is clustered toward the origin. We would suspect then that the MVE would be in this area also. Figure 3 shows the data set that corresponds to the true MVE as given in Hawkins [2]. As expected, the MVE is near the origin. When the EID method is applied to this data set, the first observations that are deleted are the outlying ones in the upper right-hand corner of the plot. It is not until the last points are deleted that the EID algorithm makes an incorrect choice. The set chosen by the EID approach is shown in Figure 4. Note the point that is incorrectly retained in the set. One reason for this error is that the point the EID deletes has a larger magnitude than the one that should be kept in the set. As stated before, these will be the points that tend to have a larger EID value in some cases.

Finally, one last comparison is in order regarding the 'salinity' data set. It is stated in Hawkins [2] that this set would require approximately 5,000 random starts with the FSA to reliably determine the MVE which is a computationally intensive task. Note that for this data set the EID method of subset selection finds a set of points in 0.22 sec with only 3% error in the volume of the ellipse.

5. Summary

In this paper, the EID method of determining the subset of points used in the MVE has been described. Subset selection is what makes the MVE a computationally expensive algorithm to implement in daily practice. Preliminary results indicate that the EID method for selecting the set of points to be included in the MVE estimator is a useful one. The time required for subset selection is less than a second for the data sets considered here, and it is expected that for large n similar savings in time can be achieved.

The 2-dimensional scatterplots of the 'delivery' data indicate qualitatively that the EID tends to pick a tighter cluster of points. Whereas the set of points making up the true MVE is somewhat narrower. This example helps illustrate an important point about the MVE. Since it is an ellipsoid of minimum volume it does not necessarily pick the tightest cluster of data. It is suspected that the EID approach might yield better results based on some other criterion; e.g., better covariance structure or clustering. These ideas will be examined in more detail as part of the future work in this area.

Although the EID method is not guaranteed to find the true MVE, it has certain advantages that make it more attractive than the algorithms currently in use. As discussed previously, it involves little computational effort, and thus it is suitable for sets with large n and p. Also, due to the iterative nature of the method, it would be easy to get a family of estimators for different values of h which is a useful feature [2]. This makes the use of the EID method feasible, thus allowing the statistician to easily employ this robust method of estimating multivariate location and scatter.

Acknowledgments

The authors would like to acknowledge the support of the Independent Research Program, Naval Surface Warfare Center, Dahlgren Division.

References

[1] P. J Rousseeuw, (1985), "Multivariate estimation with high breakdown point," *Mathematical Statistics and Applications*, ed. by Grossman W., et. al., Reidel Publishing, Dordrecht.

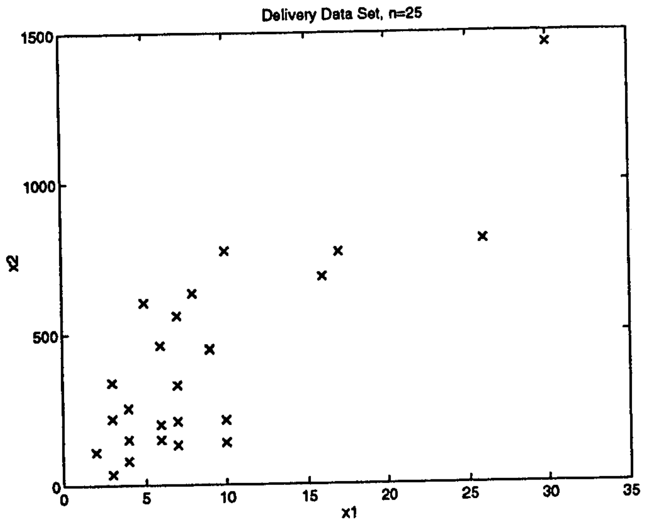


Figure 2. Scatterplot of entire 'delivery' data set.

[2] D. M. Hawkins, (1993), "A feasible solution for the minimum volume ellipsoid estimator in multivariate data," *Computational Statistics*, p. 95.

[3] P. J. Rousseeuw and A. M. Leroy, (1987), *Robust Regression and Outlier Detection*, John Wiley & Sons, New York.

[4] D. L. Woodruff and D. M. Rocke, (1993), "Heuristic search algorithms for the minimum volume ellipsoid," *Journal of Computational Statistics and Graphics*, Vol. 2, No. 1, p 69.

[5] D. M. Titterington, (1975), "Optimal design: some geometrical aspects of D-optimality," *Biometrika*, Vol. 62, No. 2, p 313.

[6] D. C. Kammer, (1991), "Sensor placement for on-orbit modal identification and correlation of large space structures," *Journal of Guidance, Control and Dynamics*, p 251.

[7] W. L. Poston, (1991), "Optimal sensor locations for on-orbit modal identification of large space structures," Master's Thesis, George Washington Univ.

[8] W. L. Poston, C. E. Priebe, and O. T. Holland, (1995), "Maximizing the fisher information matrix in discrete-time systems," *Digital Design and Control Systems*, ed. C. Leondes, Academic Press.

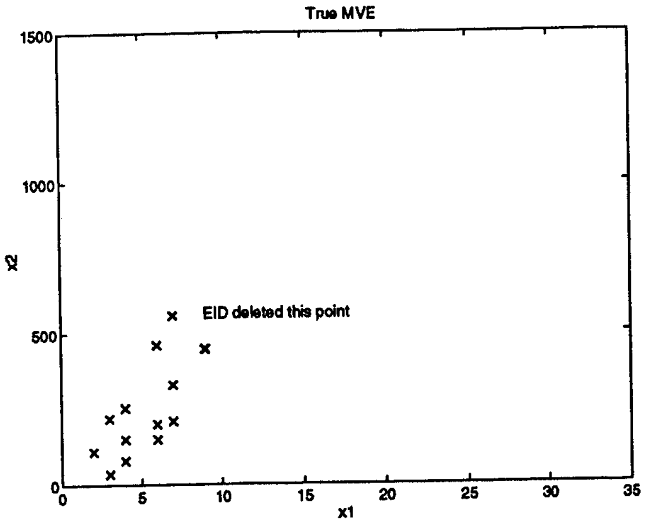


Figure 3. Scatterplot of the *h* points that are covered by the true MVE. Note the point that is incorrectly deleted by the EID algorithm.

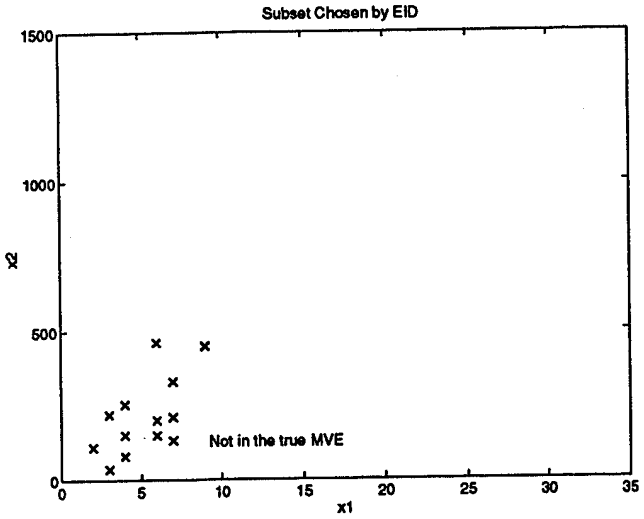


Figure 4. Scatterplot of the *h* points chosen by the EID method. Note the point that should have been deleted.

[9] W. L. Poston and R. H. Tolson, (1992), "Maximizing the determinant of the information matrix with the effective independence distribution method," *Journal of Guidance, Control, and Dynamics*, p 1513.

COMPUTING NONPARAMETRIC FUNCTION ESTIMATES

Clive R. Loader
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974, USA

Abstract: We discuss computational approaches to multivariate function estimation using strengths from the spline and local fitting worlds. In particular, we consider piecewise polynomial representations of the surface that can be easily evaluated at any point in the domain, and estimating parameters using local approaches. Ideas along these lines using rectangular partitions of the domain have been used previously by Silverman (1981) and Cleveland and Grosse (1991). In this paper we propose triangular partitions which may offer greater efficiency and adaptability in modeling surfaces.

1 Introduction

Figure 1 is a contour plot of a density estimate. The sample here consists of 225 observations from an equal mixture of three standard bivariate normal distributions.

This figure was constructed by direct application of the local likelihood density estimate, discussed in Loader (1993). In particular, for each point x on a 100×100 grid, the following system of equations was solved for a:

$$\frac{1}{n}\sum_{i=1}^{n} A(X_i - x)K\left(\frac{X_i - x}{h}\right)$$
$$= \int A(u-x)K\left(\frac{u-x}{h}\right)e^{\langle a, A(u-x)\rangle}du \qquad (1)$$

Here, $A(v)$ is a vector of the ten cubic basis polynomials and a is a vector of unknown local coefficients. The density estimate is $\hat{f}(x) = \exp(\langle a, A(0)\rangle)$. As an estimate, the local likelihood method performs quite well; the three peaks are clearly separated and are reproduced almost to full height; the true peak height is about 0.053.

However, as a computational method direct application of the local likelihood method is wasteful and inefficient. The local likelihood method assumes the underlying density is smooth, and therefore it seems unnecessary to work independently at very close points.

Moreover, we only have the estimate evaluated on a discrete grid of points; for many purposes, this may be unsatisfactory. To use the fitted surface for classification or prediction requires us to be able to readily evaluate the surface at any point; not just the grid points. In the regression setting, a residual plot requires evaluation of the surface at the data points. Particularly in higher dimensions, one must visualize the surface by looking at lower dimensional sections. Or, one may be interested in looking more closely at part of the surface.

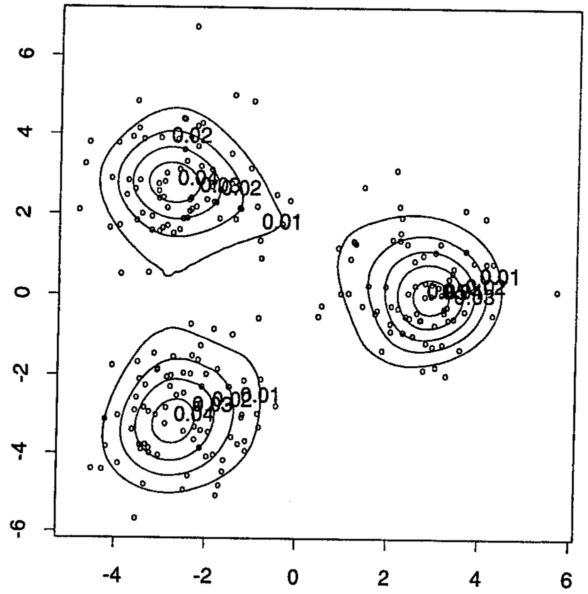


Figure 1: Density Estimate from 225 points. Direct application of the local likelihood method. Local cubic fitting was used with a variable bandwidth covering 113 data points.

The subject of this paper will be better ways to compute local fits. In particular, we require efficient ways to compute and represent the fitted surface. The examples presented are all density estimation, although very little is specific to this case.

2 What Computation is, and is not

The computation of local fits is often portrayed in the statistics literature as a race to evaluate the surface as fast as possible at a predetermined set of points. This represents a very narrow focus; while computation speed is certainly important, there are many other important considerations.

Fundamentally, the fit must summarize the data; not vice versa. The surface should be represented in a compact manner to allow prediction, classification, resampling, interactive visualization etc. Just computing the surface on a grid of points is in general not satisfactory.

An algorithm for local fitting generally consists of a number building blocks. On top of basic fast algorithms, one may want additional features, such as choices in the order of fitting; robustness; variable bandwidths and application to a wide variety of settings such as regression, density estimation, local likelihood fitting and multivariate problems. Ideally, the basic computational algorithms should be as flexible to as many of these settings as possible. Methods that require a fixed bandwidth, or only work in one dimension, are of very limited utility.

For the statistician, it is important that procedures be implemented within the environments with which they are familiar, and that provide facilities for data management and interactive analysis and graphics and the like. A stand-alone C or Fortran program will certainly be much faster to execute than software incorporated in a statistical computing environment; however it is much less useful to the statistician because of the overhead required to manage data, interface with other systems for display etc.

We can now state why some of the 'fast' methods discussed in the statistics literature are essentially useless as general purpose computational algorithms. As particular examples, we focus on binning and updating methods, presented for example by Fan and Marron (1994). First we note the comparisons in that paper are essentially meaningless in light of the preceding discussion: Figure 8 of Fan and Marron (1994) compares times for local linear smoothing; one curve for a weighted smoother with robustness iterations running in the S environment while the methods promoted by Fan and Marron use a uniform kernel smoother without robustness iterations and compiled as a stand-alone C program. This comparison says nothing about the basic computational algorithms.

The binning method divides the predictor space into intervals or a grid, and assigns observations to grid points. For a large samples, this effectively reduces the sample size, thereby speeding up the computations. But the binning step is essentially a preliminary local constant fit; to avoid bias problems, large numbers of bins are required to avoid bias problems. This is particularly so in the multivariate case. Fan and Marron report the major speedups result from reducing the observations to an equally spaced grid, reducing the number of kernel evaluations. However, this only works with a constant bandwidth!

Updating methods are designed for polynomial weight functions, and expand sums such as those on the left hand side of (1) in powers of the X_i. The sums are then updated by moving to nearby x values. However, currently available implementations only address one dimensional problems, and the fastest implementations have serious stability problems.

Most importantly, both of these methods require solution of the optimization problem (1) for each fitting point. Hence, particularly for our density estimation setting, these methods are unlikely to be particularly fast. Also, the methods are geared towards evaluating the estimate on a predetermined set of points (for binning, on an equally spaced grid), which is not satisfactory for many statistical purposes such as prediction, classification, resampling and interactive graphics. Neither method provides a compact representation of the fitted surface.

3 Piecewise Polynomials

Suppose we have a sequence of vertices $v_0 < v_1 < \ldots < v_k$, and evaluations of a function $f(v_i)$ and its derivative $f'(v_i)$ at each vertex. There exists a unique C^1 function that matches the function values and derivatives at each vertex, and is piecewise cubic on each interval (v_i, v_{i+1}). This type of construction enables us to approximate smooth functions quite cheaply using Hermite polynomials. See De Boor (1978, Chapter 4) for further discussion of this scheme.

Another closely related method is the cubic spline approximation. In the most common use, the cubic spline method enforces continuity of the first and second derivatives; it is not required to match the true derivatives.

Several methods along these lines have been used in nonparametric function estimation. Penalized likelihood methods (Wahba, 1990) give rise to a cubic spline estimate with vertices at the data points. Regression splines, and the logspline density methods of Kooperberg and Stone (1991) estimate the parameters of a spline using criteria such as maximum likelihood; this enables the number of vertices to be substantially reduced. An alternative method is to estimate the parameters using local regression or likelihood methods. For this approach to be computationally efficient, it is important to make effective use of the vertices; for this reason the Hermite polynomial rather than cubic spline approach is preferable. This is the idea underlying the LOESS method (Cleveland and Grosse, 1991).

The computational advantage of local regression and likelihood appears in multidimensional cases. Global approaches to fitting generally involve the solution of a large scale optimization problem, which is generally very expensive. While sparse matrix techniques have been successfully applied to spline problems in one dimension, there use in multiple dimensions is much more difficult. By comparison, local regression methods solve a small optimization problem at each vertex.

Two fundamental problems remain: The construction of a suitable partition of a multidimensional domain, and the construction of interpolants over this partition. The most common types of partition involve rectangular cells or triangular cells.

A rectangular partition is often preferred in theoretical work, since it is often easier (or less difficult) to analyze, and define suitable polynomial approximations. The earliest work using local fits in this direction appears to be Silverman (1981) who constructed an estimate based on evaluation of a fixed bandwidth kernel estimate and its derivatives over a coarse grid, and used a piecewise quadratic scheme to interpolate over the cells of the grid.

A further development in this direction is the k-d tree structure introduced by Friedman, Bentley and Finkel (1977) and applied to local regression by Cleveland and Grosse (1991). In this algorithm, the data is initially bounded by a box, and the cells are recursively split. The splits in the k-d tree algorithm always divide observations in the parent cell into two subsets of approximately equal size, and hence the resultant structure has most vertices in regions of high point density. This is commensurate with the nearest neighbor bandwidths used in LOESS; however, other split rules could be used to adapt to other situations.

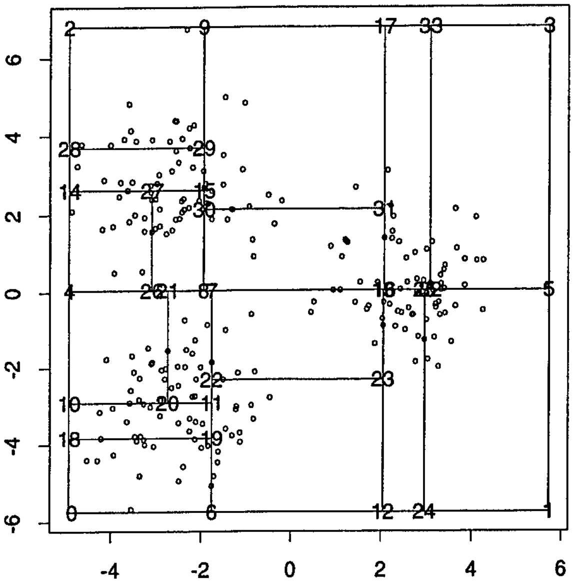


Figure 2: Division of 225 data points by a k-d tree with 34 vertices and 16 cells. Vertices are numbered in the order they are entered into the tree.

An example of a k-d tree is shown in Figure 2. As required, this partition has most vertices in regions of high density. One can also identify some weaknesses; in particular, occasionally vertices occur at very close points suggesting inefficiency. Also, construction of interpolants over a rectangle depends on more than just the corners; for example, interpolating over the rectangle $(10, 11, 18, 19)$ cannot ignore the evaluation at vertex 20. This problem becomes particularly complex in three or more dimensions.

An alternative to rectangular cells is partitions based on triangular cells. A triangle in d dimensions has $d + 1$ vertices, against 2^d for cubes. This suggests there are *potential* savings in multiple dimensions. The word potential is stressed here, since there are both 'good' and 'bad' triangles.

Some examples of good and bad triangles are shown in Figure 3. The ideal triangulation would consist of equilateral triangles. In two dimensions, these can be tessalated; unfortunately this is non-adaptive and therefore may be wasteful in regions of low density where a large bandwidth must be used. The right angled triangle scores 'ok'; this triangle can be interpolated over with reasonable success, but we are unlikely to gain much over the use of rectangular grids. The tall isosceles triangle is poor, since some points may be far from the nearest vertex of the triangulation. Finally, the flat isosceles triangle scores bad; interpolating in the middle of the vertical edge will ignore the two side vertices.

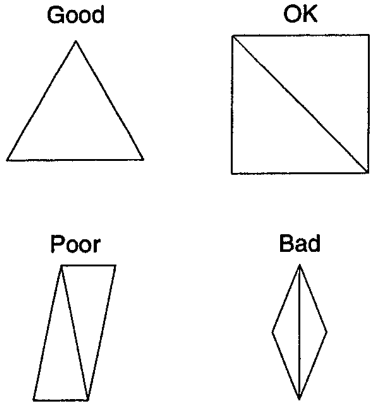


Figure 3: Examples of good and bad triangles in a triangulation.

In addition to requiring good triangles, there are several other competing criteria that must be considered when growing a triangulation. We wish to maintain adaptiveness; in particular, there should be more vertices in regions where small bandwidths are used. The triangles must be small enough for interpolants to work reasonably; however, the vertices should be sparse enough for efficiency. Finally, the triangulation must be fast to grow and search.

4 Recursive Partitioning

In this section, one possible triangulation scheme based on recursive partitioning is presented. First, draw a box around the data, and divide the box into two triangles by the addition of a diagonal. The triangulation is then grown recursively by adding vertices to existing edges. Suppose existing vertices are $v_1, \ldots, v_m$, and $h_1, \ldots, h_m$ are the bandwidths used at these vertices. For each edge (v_i, v_j) in the existing triangulation, assign a score:

$$\rho_{i,j} = \frac{\|v_i - v_j\|}{\min(h_i, h_j)}.$$

If $\rho_{i,j} > c$ split the edge. Let $\lambda = h_i/(h_i + h_j)$, and create a new vertex at $v_{m+1} = \lambda v_j + (1 - \lambda)v_i$. We then create the new edges joining points to v_{m+1}. This process is repeated until no more edges require splitting.

The use of bandwidths in determining the splits allows the algorithm to preserve adaptivity. If h_i is smaller than h_j,, then λ will be less than 0.5 and v_{m+1} is closer to v_i. In extreme cases this can over adapt, and so we restrict λ to the interval (0.2, 0.8).

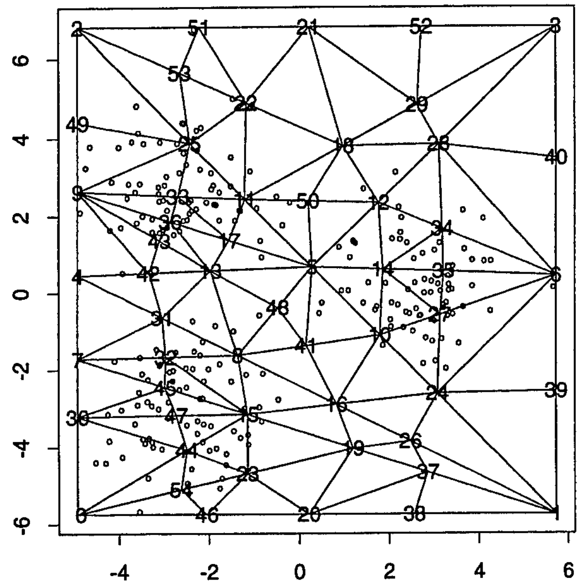


Figure 4: Recursive triangulation. Vertices are numbered in the order they were entered into the triangulation.

Figure 4 shows a recursive triangulation grown on the trimodal data in Figure 1. The vertices are numbered in the order they were entered into the triangulation. The bandwidth used is variable, covering 113 nearest neighbors for each fitting point. Some adaptivity can be seen in this picture: There is a greater density of vertices around the three peaks, where a smaller bandwidth is used for the fit.

The recursive partitioning presented here has both good and bad features. A big advantage is the triangulation can be stored in a tree-like structure; for any point x, it can then be rapidly determined which triangle contains x. The disadvantage is a recursive scheme scores only an 'OK' in terms of goodness of triangles, and so one can't expect substantial gains in efficiency. There are also some bad splits in the algorithm; for example, point 7 was used to split the triangle (0, 1, 4); it would have been better for the algorithm to split the (1, 4) edge first.

5 Finite Element Interpolants

The finite element method constructs interpolants over the cells of a partition. Only vertices on the boundary of a cell are used. This has computational advantages; the interpolants depend on only a small number of parameters and hence do not require solving large systems of equations.

The cell-based construction of finite elements leads to substantial difficulty in enforcing global smoothness conditions. For visualization purposes, we would like our surface to be continuous and differentiable. Also, the interpolant should be commensurate with our fitting procedure; for example, an interpolant that only reproduces linear polynomials is not adequate for use with a local quadratic or cubic fitting procedure.

A two dimensional element suitable for our purposes is the Clough-Tocher finite element (Clough and Tocher, 1965; Lancaster and Salkauskas, 1986). This method uses twelve pieces of information: The function values and derivatives at each vertex of the triangle, and the normal derivatives at the midpoint of each side. The Clough-Tocher finite element is then piecewise cubic over each of three sub-triangles, with continuity and differentiability conditions enforced at the interior seams. See Figure 5. A remarkable feature of the Clough-Tocher method is it produces a *globally* C^1 surface; when the method is applied independently on adjacent triangles, the resulting surface is differentiable at the common boundary!

The full twelve parameter Clough-Tocher method will reproduce a cubic polynomial. Unfortunately our local fitting procedure will not produce the normal derivatives at the midpoints of the sides; in practice, these are estimated by linear interpolation. This reduced nine parameter Clough-Tocher method reproduces all quadratic terms. A cubic reproducing scheme could be constructed by using second derivatives at the vertices and estimating normal derivatives using quadratic interpolation.

Figure 6 shows the Clough-Tocher method applied to the data from Figure 1 using the triangulation in Figure 4. Qualitatively the picture looks very similar to the direct fit; the three peaks are kept separate and reproduced

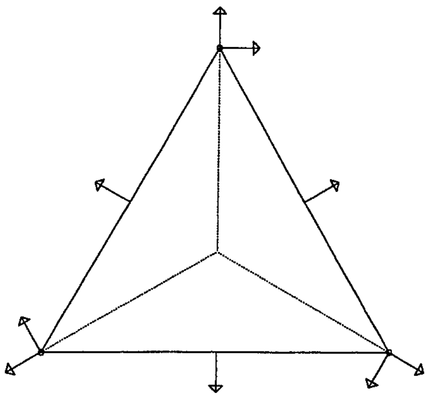


Figure 5: Clough-Tocher method. The interpolant uses function values at the three vertices of the triangle, and the nine directional derivatives indicated. The interpolant is made up of three piecewise cubics over the interior triangles.

to a similar height in both figures. Some differences are visible, particularly in the 0.01 contours.

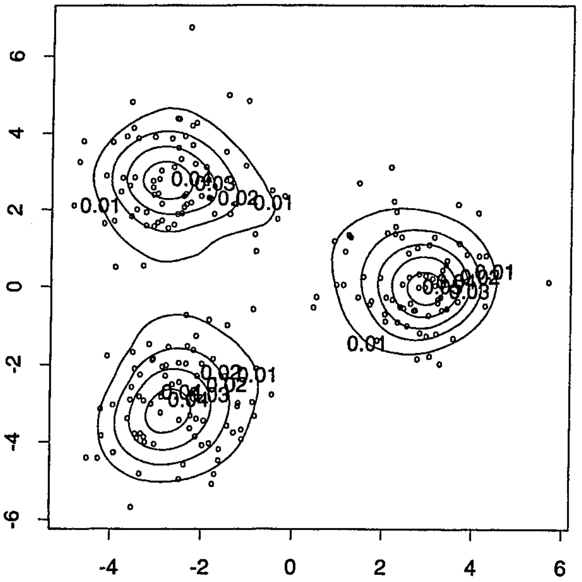


Figure 6: Density estimate using Clough-Tocher interpolation over a triangulation. Local cubic fitting.

	No. Vertices	$d(\tilde{f},\hat{f})$	$d(\tilde{f},f)$
Direct	225	-	0.261
Triangulation	55	0.0296	0.269
k-d tree	34	0.0638	0.282
k-d tree	66	0.0345	0.264

Table 1: Comparison of direct fitting with triangulation and k-d tree based interpolation schemes.

Alternative constructions of interpolants can be based on piecewise quadratic, rather than cubic, polynomials. This has advantages noted by Silverman (1981) for producing contour plots since level sets can be readily found. Another advantage is in locating local maxima of the estimate. The difficulty is that a quadratic has fewer degrees of freedom, and additional internal seams must be introduced to enforce boundary constraints. For example, Silverman divides each cell into sixteen triangles. For a construction of a C^1 quadratic surface on a triangular partitions, see section 6.2 of Chui (1988).

6 Concluding Remarks

The triangulation approach is competing with the k-d tree as an approximation method. A natural question is to try to make an objective comparison of their performance. We consider 'comparable' estimates to consist of the same or similar numbers of vertices. The real question of course is how well do the interpolated estimates approximate the true estimate. In practice we cannot measure this, so we also consider how well the interpolated estimate approximates the direct. We consider the criterion

$$d(\tilde{f},\hat{f}) = \frac{1}{n}\sum_{i=1}^{n} |\log \tilde{f}(X_i) - \log \hat{f}(X_i)|$$

where $\hat{f}(x)$ is the direct density estimate and $\tilde{f}(x)$ is the interpolated density estimate. where $\hat{f}$ is either density or log-density; $\hat{f}_0$ is direct and $\hat{f}_1$ is triangulation or kdtree estimate. This is also an approximation to L_1 distance $\int |\tilde{f}(x) - \hat{f}(x)|dx$ between the two estimates.

Table 1 shows the results for our triangulation and two different sizes of k-d tree. As we would hope, the 34 vertex k-d tree is substantially beaten. The triangulation also slightly beats the 66 vertex k-d tree compared to the direct estimate, but loses slightly compared to the true density. Of course, not too much should be concluded from one example selected by the author; however we believe this certainly gives grounds for optimism.

The triangulation used in section 4 is based on recursive partitioning. This enables the triangulation to be stored as a tree type structure, which facilitates rapid searching; in particular, one can rapidly determine which triangle contains an arbitrary point in the domain. There are however some disadvantages; in particular, we have noted the resulting triangulation will generally only score 'OK' on the scale of Figure 3. We have also given up another major advantage of triangulations; namely the ability to well approximate fairly arbitrary nonlinear boundaries and domains.

There are alternative methods of growing triangulation, any of which may have advantages and disadvantages in our application. An obvious alternative is a sequential scheme, beginning with a seed triangle in the center of the data, and adding neighboring triangles until the domain is filled. This loses the tree structure of our recursive scheme. Some advantages are that 'good' triangulations may result more readily, and easier adaptability to non-rectangular boundaries.

Another step that might be considered is to attempt to optimize the triangulation in some sense. While a full blown optimization is the type of expensive problem we seek to avoid, some improvement may be obtained by moving vertices around or constructing triangulations on a given set of vertices. Some discussion of such schemes can be found in Barnhill (1977).

There are a number of directions in which this work can be extended. Perhaps the most obvious is beyond two dimensions, where as already noted there is potential for triangulations to be much more efficient than rectangular grids.

An obvious extension of this work (indeed the original motivation) is to higher dimensions, where the potential saving of triangular cells over rectangular by reducing the number of vertices seems much greater. Of course, any nonparametric fitting becomes much more difficult beyond two dimensions due to data sparseness. However, the construction of good triangulations is more difficult; also difficult is the problem of enforcing global differentiability in finite element interpolants.

There are also some theoretical questions that could be pursued. One obvious question is to analyze how well interpolated methods perform as estimates; in particular, we hope to preserve good properties of local polynomial fitting. Such an analysis may also help suggest refinements to the triangulation; in particular, how deep should the triangulation be taken? Another question is how to estimate derivatives. The present implementation uses the local slopes from the local polynomial fits; this is convenient since the slopes are available at no extra computation costs. Theoretical analysis suggests these slopes have good properties as derivative estimates; however, not always at the same bandwidth as for estimating the function itself. The author has found this problem severe in some cases for local quadratic fitting; hence the preference for local cubic fitting in the examples in this paper.

Finally, we mention some ongoing work in nonparametric function estimation related to the topics discussed in this paper. Eric Grosse is considering alternative constructions of Loess estimates based on the k-d tree partition; in particular, placing vertices at the centers of the cells, rather than the corners. This solves the problem of nearby vertices mentioned in relation to Figure 2, and should improve efficiency of the scheme. Mark Hansen and Charles Kooperberg are using triangulation methods in a global likelihood estimation scheme, considering generalized vertex splines (Chui, 1988) and other methods to construct estimates.

REFERENCES

BARNHILL, R. E. (1977). Representation and approximation of surfaces. In *Mathematical Software III*, Ed. J. Rice. Academic Press, New York.

CHUI, C. K. (1988). *Multivariate Splines.* SIAM, Philadelphia.

CLEVELAND, W. S. and GROSSE, E. (1991). Computational methods for for local regression. *Statistics and Computing* **1**, 47-62.

CLOUGH, R. W. and TOCHER, J. L. (1965). Finite element stiffness matrices for analysis of plates in bending. *Proc. Conf. Matrix Methods in Structural Mechanics*, Wright-Patterson A.F.B., Ohio.

DE BOOR, C. (1978). *A Practical Guide to Splines.* Springer Verlag, New York.

FAN, J. and MARRON, J. S. (1994). Fast implementations of nonparametric curve estimators. *J. Comp. & Graph. Statist.* **3**, 35-56.

FRIEDMAN, J. H., BENTLEY, J. L. and FINKEL, R. A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software* **3**, 209-226.

KOOPERBERG, C. and STONE, C. J. (1991). A study of logspline density estimation. *Comp. Statist. & Data Anal.* **12**, 327-347.

LANCASTER, P. and SALKAUSKAS, K. (1986). *Curve and Surface Fitting: An Introduction.* Academic Press, London.

LOADER, C. (1993). Local Likelihood Density Estimation. Available by ftp from `netlib.att.com` in the file `netlib/att/stat/doc/93/31.ps.Z`.

SILVERMAN, B. W. (1981). Density Estimation for Univariate and Bivariate Data. In *Interpreting Multivariate Data*, 37-53. Ed. V. Barnett. Jon Wiley & Sons, Chichester.

WAHBA, G. (1990). *Spline Models for Observational Data.* SIAM, Philadelphia.

DATA CONVERSION PITFALLS

Terry J. Woodfield
Risk Data Corporation
111 Pacifica, 3rd Floor
Irvine, CA 92718-3331

Abstract

Electronic bulletin boards, client/server computing, and inter-network connectivity have promoted the transfer and exchange of data. Differences in computer hardware prevent seamless transfer of data from one computer system to another. Data transfer implies data conversion except when compatible hardware is employed. Even when compatible hardware is used, careless specification of data formats may lead to serious problems. Situations arise where improper conversion may go undetected, leading to the statistical analysis of invalid data. This article addresses data conversion issues and highlights pitfalls that impact the transfer of numerical data between different computer systems.

1. INTRODUCTION

An insurance company must deliver an extract of claims data stored on an IBM mainframe computer to a small information services company which operates a network of UNIX workstations and personal computers. The insurance company provides the data on 3480 tapes using a variable length record layout. A block size B is specified, and records have no record delimiters. Instead, individual records are determined by processing the first four bytes of a record as the record length in bytes. Blocking is determined by a four byte field that begins each block and gives the block size in bytes. A block may contain more than one record, but no partial records. Both block size and record length information is stored in binary.

Fields in the data extract are copied directly from the claims system, so the extracted data contains a variety of data types, including a numeric format called "packed decimal" which will be described later. Furthermore, no conversion is applied to the extracted data, so character data is stored using EBCDIC encoding.

On the information services company side, the UNIX and PC systems use ASCII encoding. The systems administrator reads the tape dataset directly onto the hard disk of a UNIX workstation. The blocks are converted from EBCDIC to ASCII before writing to disk. The systems administrator reasons that data from an EBCDIC system must be converted to ASCII for the ASCII-based UNIX system.

The statistician obtains a 66,077,850 byte file (approximately 63 Mb) with the above information, a detailed record layout, and block size $B = 32000$. She reads the first four bytes using a positive integer binary format and obtains the value 16,459, which she interprets to be the size of the first block. The documentation implies that record sizes are one of 260, 270, 280, or 290 bytes. An odd block size is not possible even with the four byte block header.

The first 20 fields on all records are the same, and the fifth field is a four byte packed decimal field. Her attempts to read the fifth field of the first record as a packed decimal field fail. She gets the hex value 0003A61Ch. The statistician is now convinced that the data is corrupt.

In this example, the statistician received the data indirectly from a DP staff member. She was informed by the person who transferred the data from tape to disk, "I read the data and converted it using the system `dd` command with conversion equal ASCII turned on." At the time, it made sense to convert IBM 370 data created on an EBCDIC system to the standard ASCII encoding used on UNIX workstations. This seemingly insignificant conversion exercise effectively encoded the data into an unusable but salvageable form. Had the data remained in "EBCDIC" format, the block header would have been read as a realistic value of 31954, and the packed decimal field would have produced 0003471Ch, or 3,471.

With the rise of client/server computing and networks of computers exchanging and sharing information, statisticians must be aware of the potential pitfalls inherent in the transfer and conversion of data. This paper addresses some of the pitfalls and provides guidance for transferring data and diagnosing problems.

2. EBCDIC AND ASCII ENCODING

The ASCII (American National Standard Code for Information Interchange) conversion standard associates a one byte binary integer value with an interpreted 'character', which can be: (1) a printable character, referred to as an ASCII character, (2) a communication or 'handshaking' code, or (3) an 'escape' or 'control' character. ASCII decimal values 32 through 126 (hex 20h through 7Eh) are represented by standard keyboard symbols and are often referred to as 'printable characters'. These include symbols like !, <, 9, A, a, and]. If Shift-A, is typed, 'A' is displayed, and the decimal value 65 (hex 41h) is placed in the keyboard input queue. ASCII decimal values 0 through 31 (hex 0h through 1Fh) have meanings that may vary depending on the software used or the device that is sending or receiving the values. For example, the linefeed character is ASCII decimal 10 (hex Ah), form feed is ASCII decimal 12 (hex Ch), and carriage return is ASCII decimal 13 (hex Dh). ASCII decimal 3 is the familiar Control-C. ASCII decimal values 128 through 255 may have standard meanings, but many software products do not recognize the standard values for 'extended ASCII'.

The EBCDIC (Extended Binary-Coded Decimal Interchange Code) conversion standard performs the same function as ASCII, except EBCDIC has a richer set of printable characters, including characters ¢ and ¬.

ASCII and EBCDIC imply a sort order for symbols, and the order is different. For example, EBCDIC sort order places b (decimal 130) before B (decimal 194), whereas ASCII sort order places B (decimal 66) before b (decimal 98).

If data is stored as EBCDIC, it may be converted to ASCII, although conversions for extended ASCII or for non-printable characters may be dependent on the application. The UNIX dd command employs a one-to-one mapping between the two encodings. Some software products do not provide a one-to-one mapping for ASCII codes above 127. For non-standard encodings, such as ASCII 128 or EBCDIC 112, conversion from one encoding to another is risky, and data with non-standard encodings should be investigated. Most data represents printable characters or numeric values. Character data with non-standard encoding should be suspect.

When data is not encoded as pure ASCII or pure EBCDIC, then ASCII or EBCDIC conversions must be avoided. Packed decimal is neither ASCII nor EBCDIC. Numeric formats are more hardware dependent than are character formats like ASCII and EBCDIC. Two UNIX workstations may use ASCII encoding, but may use totally different binary or floating point storage. The mistake that computer novices make is to assume an entire dataset is ASCII or EBCDIC, when in fact only certain data fields within records are ASCII or EBCDIC.

3. NUMERIC FORMATS

The two most common numeric formats are integer binary and floating point binary. While some systems may store integer values in reverse bit order or vary where the sign bit is located, many systems use a direct binary translation of integer values. Thus, the rightmost bit represents units, the next bit two's, the next bit four's, etc.

Floating point storage is another matter, with system 370 format and IEEE format dominating. Single precision and double precision are the two standard floating point storage sizes. All floating point storage modes involve a *mantissa* that records all significant digits, and an *exponent* which defines the location of the decimal point. Floating point formats differ in: (1) the size of the mantissa and exponent, and (2) the method of storing the numeric information.

There are 4,278,190,592 valid floating point values using the IEEE four byte (single precision) floating point storage format. There are 16,776,704 'invalid' floating point values using the same IEEE storage format. These invalid values may actually be interpreted as 'infinity' or 'not-a-number' (NAN) values. If you randomly generate four byte values using a uniform generator, approximately 99.61% of the values should be valid floating point representations.

Since floating point is so dependent on hardware platform, it is rarely a good storage format for transferring data across systems. For this reason, numeric storage modes like packed decimal and zoned decimal are popular.

Packed decimal interprets the hex representation of a number as a decimal number. The last half-byte is hex C for positive values and hex D for negative values. Thus, 973Ch is positive 973, and 450Dh is negative 450.

Zoned decimal uses a single byte for each numeric digit. Fixed point storage is assumed with an implied decimal. The sign, plus or minus, is stored in the last byte, and the nature of the sign byte depends on whether the system is ASCII or EBCDIC. If the rightmost digit is 0, and the number is positive, the rightmost byte will have value '{' (EBCDIC hex C0) or '0' (ASCII hex 30). If the rightmost digit is 0, and the number is negative, the rightmost byte will have value '}' (EBCDIC hex D0) or 'p' (ASCII hex 70). Positive sign bytes progress from hex C0 to hex C9 on EBCDIC systems, and from hex 30 to hex 39 on ASCII systems. Negative sign bytes progress from hex D0 to hex D9 on EBCDIC systems, and from hex 70 to hex 79 on ASCII systems. The rightmost hex digit is the value of the last digit in the number.

The following table presents statistics on an arbitrary set of text files read in as 4 byte floating point words using IEEE single precision floating point storage.

Text Files Read as Floats, 4 Byte Words			
Source	Mean	Minimum	Maximum
IEEE value	3.047e296	2.646e-260	1.367e301
IEEE exponent	161.85	-260	301
S370 value	7.475e70	9.658e-67	2.057e74
S370 exponent	39.17	-67	74

The unusual magnitude of the values should be sufficient to indicate that the wrong conversion format was employed. Large ranges imply that the floating point format is invalid.

4. DATA CONVERSION EXAMPLES

The conversion of data written on one computer to a form acceptable to another computer may appear to be straightforward, but problems may arise. If a record from a dataset contains non-numeric data in some fields and numeric data stored in binary or some other form in other fields, then a simple conversion of the entire record is not possible. A common situation that is often encountered is the conversion of EBCDIC to ASCII. For such situations, the statistician must distinguish between fields that contain EBCDIC values and fields that contain numeric values. A common misconception is that numeric values expanded into EBCDIC symbols are stored as numeric values. An example serves to illustrate this problem. In the following, the symbol '@' is used to represent an unprintable character.

```
Record Layout:
  Field  Columns  Format   Value
   NAME     1-20  EBCDIC   JONES, ULYSSUS P.
 AMOUNT    21-24  EBCDIC   $215
   LOSS    25-28  Binary   $19,608

                           1    1    2    2
   Column Number: ----5----0----5----0----5---
                  DDDCE64EDEEEEE4D4444FFFF0049
     Record in Hex: 16552B0438224207B000021500C8
Record in EBCDIC: JONES, ULYSSUS P.   0215@@<q
```

Note that the display stacks hex digits so that the relationship between what is seen and the corresponding hex representation is clear. For example, the character J is hex D1, O is hex D6, etc. The digits 0 through 9 are represented in EBCDIC hex as F0 through F9.

In the example, AMOUNT is stored in EBCDIC rather than integer binary. This is wasteful, because amounts between $0 and $10,000 may be stored in binary using two bytes rather than four bytes, because

$$9999(\text{decimal}) = 0010\ 0111\ 0000\ 1111(\text{binary}).$$

Nonetheless, it is convenient to store data in readable form when small numeric quantities are involved. The field LOSS is stored in integer binary using four bytes, which permits dollar amounts up to $2,147,483,647. Had EBCDIC been used, ten bytes would have been required. For values up to $8,388,607, only three bytes are required.

For the example, note that

$$\begin{aligned} 19,608(\text{decimal}) &= 4C98(\text{hex}) \\ &= 0100\ 1100\ 1001\ 1000(\text{binary}). \end{aligned}$$

If the four byte LOSS field was interpreted as EBCDIC, then the value would be treated as invalid. The representation "@@<q" cannot be interpreted as a number. On the other hand, if the four byte AMOUNT field had been interpreted as integer binary, the amount would have been read as $4,042,453,493. The lesson is that any data field interpreted as integer binary will produce valid numeric values, but EBCDIC fields may contain values that cannot be interpreted as numeric.

The most serious consequence of poor conversion is that integer binary fields may be converted from EBCDIC to ASCII, thereby producing bogus results. For example, converting the above example record to ASCII produces

```
                           1    1    2    2
   Column Number: ----5----0----5----0----5---
                  4444522545555525222233330037
     Record in Hex: AFE53C05C9335300E000021500C1
Record in EBCDIC: JONES, ULYSSUS P.   0215@@<q
```

Note that the printable characters do not change, but since the LOSS field was not meant to represent printable symbols, the numeric value of LOSS has changed. In this case, LOSS takes the value

$$3C71(\text{hex}) = \$15,473(\text{decimal}).$$

For many situations, descriptive statistics will reveal the conversion mistake, but as this example illustrates, situations exist where the mistake would not be obvious. A small simulation reveals that the problem can be serious when numeric values are relatively small. The following table contains results for a simulated dataset having 500 observations taken from a Gamma distribution with shape parameter 2 and scale parameter 100. Values were written to disk as four byte integer binary values. The column headed "ASCII Value" represents the data read in after using the UNIX dd command EBCDIC-to-ASCII conversion feature, and the column headed "EBCDIC Value" represents the data read in after using the dd command ASCII-to-EBCDIC conversion feature.

Statistic	Original Value	ASCII Value	EBCDIC Value
No. Obs.	500.00	500.000	500.000
Mean	197.51	207.242	227.598
Median	159.50	180.000	208.000
Std Dev	141.78	143.748	142.722
Min	6.00	4.000	5.000
Max	1013.00	897.000	876.000
Skewness	1.40	1.121	1.143
Kurtosis	3.13	1.761	1.682
5%-ile	36.50	26.000	52.500
10%-ile	52.50	42.000	79.500
25%-ile	94.50	103.000	117.000
75%-ile	275.50	274.000	293.500
90%-ile	391.50	421.500	461.000
95%-ile	472.50	461.500	496.000

Many programmers who have used the built-in features of programming languages are not aware of the implications of performing arithmetic operations or storing numeric values. In fact, many databases store numeric values in human readable form. The interaction with "user-friendly" software or hardware can lead to problems or misconceptions.

A subtle data conversion problem occurs when data fields contain codes that may have special meaning to data processing software. For example, if a software product is expecting to find a record delimiter, like a linefeed (ASCII hex 0A) or carriage return (ASCII hex 0D), the software may encounter the delimiter in a packed decimal, integer binary, or floating point field. If a record delimiter is encountered within a field, the software may assume that the end of the record has been reached prematurely and truncate the record. This occurs because the algorithm reads a record into a buffer up to the record delimiter before it attempts to interpret the individual fields in the record. It may be necessary to specify record sizes rather than depend on record delimiters when special numeric storage modes are used.

When file transfer occurs using FTP or some other transfer mechanism, there are options for binary or text transfer. A fixed length record data set with numeric fields as described in the previous paragraph may be transferred as a text file. In particular, suppose the file is transferred from a DOS based system to a UNIX system. DOS terminates text records with a carriage return and linefeed, hex 0D0A. FTP replaces 0D0A with 0D, and removes the file terminating 1A (Control-Z) which DOS uses as an end-of-file marker. A binary field, say 0D0A(hex)=3338(decimal), would be left shifted one byte and replaced with 0A(hex)=10(decimal). The record containing this value would be corrupted. This explains why data files are almost always transferred as binary files.

5. DIAGNOSING BAD DATA

The most obvious approach to diagnose bad data is to flag bad values and generate a frequency table for the flag. This approach may be adequate for packed decimal, zoned decimal, or character (printable) data, but binary fields will always be valid, and floating point fields will almost always appear to be valid.

For numeric storage modes like integer binary and floating point binary, statistical summaries may be adequate. The source data set should be analyzed to derive means and percentiles. If the target dataset provides statistics that compare to within expected round-off, then it is unlikely that conversion problems occurred. Some data sources will not have expertise or resources to calculate percentiles, so one solution is to request a sum for each numeric field and a sample dump of 10 to 20 records, preferably in hex and human readable form.

What about data that comes from an "unfriendly" source? This may occur when a source is a secondhand distributor of, say, government data. The source may only be set up to distribute copies and may have no software tools to validate data. For these situations, the statistician should have some idea of what to expect from the numeric fields. A binary field will always produce valid results even if non-binary data is stored in the field, and a floating point field will appear to have valid values over 99% of the time even if the field actually contains binary or character data. On the other hand, packed decimal and zoned decimal fields that display invalid numeric values are probably specified incorrectly.

6. MISCELLANEOUS CONVERSION ISSUES

When the person writing the program to create a tape dataset has no direct interest in the statistical analysis that is to be performed, then communication problems may lead to data conversion problems. Following are a list of problems that arise from the source of the tape dataset.

- The programmer uses a system cataloged procedure without understanding the defaults that are employed.

- The programmer uses a proprietary storage mode supported only by a given commercial software product that may not be available on the system reading the data.

- The programmer anticipates EBCDIC to ASCII conversion and performs the conversion at the source cite without considering the impact on packed decimal fields or other non-EBCDIC fields.
- The programmer uses hardware specific storage modes, such as floating point and binary, and lacks the sophistication to define how these fields are interpreted on the source system.

Single purpose programmers, such as COBOL programmers performing systems analysis, may work for years without understanding how packed decimal fields are actually deciphered by the software. These programmers may also use tape utilities without understanding how they work. Tape reading and writing for archiving and other purposes is routine, but sending tapes off site may be rare. Many companies have been collecting data for years, but are just now beginning to realize the value of data. These companies are likely to be the greatest source of data conversion problems.

While only a few improper conversions are difficult to detect, a number of improper conversions may be difficult to diagnose. Integer binary being converted using EBCDIC-to-ASCII conversion tables poses a dangerous problem that is difficult to detect. Trying to read floating point data as packed decimal readily reveals a problem, but it may take some detective work to deduce the correct numeric format that should have been used.

7. CONCLUDING REMARKS

The number of seemingly sophisticated computer users who thought EBCDIC to ASCII conversion should be done for an entire dataset rather than on each individual field of a record was surprising. In a non-random sampling of associates, over 90% of those queried thought that data from an IBM system should be transferred to a UNIX system using EBCDIC-to-ASCII conversion. Rather than reflecting a basic ignorance of computer data storage, this finding probably reflects that many statisticians only use pure 'text' storage mode for data, which means that pure ASCII or EBCDIC is used.

The following guidelines for the source of the data will help make data transfer and conversion relatively painless.

1. If dataset size is not an issue, use pure EBCDIC or pure ASCII to store data.
2. Avoid using floating point storage mode on any data that is to be transferred to another computer system.
3. If system dependent storage modes like packed decimal or floating point storage are to be employed, include a complete description of the format with the dataset documentation. Complete documentation would allow the recipient to program a conversion algorithm from scratch if software tools did not exist that supported the format.
4. Include statistical summaries for all numeric fields, and frequencies for categorical fields, along with a dump of some representative records to facilitate validation on the recipients end.

Software products like the SAS System® provide tools and formats that make data transfer and conversion relatively painless. Base SAS software provides a rich collection of data conversion formats, including hardware specific formats such as IBM 370 and VAX floating point formats. The expository articles by Langston (1987) and Klenz (1992) provide insight into issues related to floating point storage of data. Kudlick (1980) is an older textbook with details about IBM System 370 numeric storage modes, such as packed decimal and floating point.

In the analysis of data obtained from a "foreign" computer source, the first step in any statistical analysis should be to verify the validity of the data. The statistician who proceeds with an analysis without having confirmed the proper transfer and conversion of data is as foolish as the scientist who comes to a statistician for help only after the data has been collected. Academicians who rely on graduate students or data center personnel should be particularly cautious, especially if their own computer skills are weak.

The SAS System is a registered trademarks of SAS Institute Inc., Cary, North Carolina.

References

[1] Klenz, Bradley W. (1992). "Handling Numeric Representation Error in SAS Applications." *Observations: The Technical Journal for SAS Software Users*, **1**, 19-30.

[2] Kudlick, Michael D. (1980). *Assembly Language Programming for the IBM Systems 360 and 370.* Dubuque, Iowa: William C. Brown Company Publishers.

[3] Langston, Richard D. (1987). "Numeric Precison Considerations in SAS Software." Proceedings of SUGI '87. Cary, North carolina: SAS Institute Inc.

Design of Object-Oriented Functions in S for Screen Display, Interface and Control of Other Programs (SAS and LaTeX), and S Programming

Richard M. Heiberger
Temple University
Philadelphia, PA 19122-2585
rmh@astro.ocis.temple.edu

Frank E. Harrell, Jr.
Duke University Medical Center
Durham, NC 27710
feh@biostat.mc.duke.edu

Abstract

We describe a set of object-oriented S functions that harness the automatic printing facility in S to convert an S object to another format. We devise new classes and subclasses of objects in S: file (with subclasses source, sas, dvi, latex, ps, and xli), display, and expr (closely related to objects of mode expression); print methods (families of related functions that depend on the class of the argument) for the new classes; and other families of functions that depend on environmental variables. We give examples for displaying an S object in its own window on the user's workstation, for converting a data.frame to a LaTeX table, for displaying S help files in a TeX window on a workstation, and for converting a data.frame to a system file for another software system. We show how to put a SAS program inside an iterative loop controlled by S. We give applications to programming and debugging in S. We discuss design issues in constructing the functions and relating the individual members of the set of functions to each other and to the object-oriented paradigm in the underlying S program.

KEY WORDS: S, Display software, LaTeX, Object-oriented programming, SAS, System Interfaces.

1. Introduction

S (Becker, Chambers, and Wilks, 1988) is a "Programming Environment for Data Analysis and Graphics" originally designed for Unix computers. S-Plus, a supported version of the program, including a port to the MS-DOS environment, is available from Statistical Sciences, Inc. (1991).

S was extended by Chambers and Hastie (1992) to include an object-oriented programming environment. Objects in the environment include functions, data, and graphs. Generic function names in the language are sensitive to the class of their arguments, and call different methods for the actual execution. For example, S normally prints quotation marks around character-valued variables but suppresses the quotation marks for character-valued columns in a data.frame.

An interactive S session consists of two types of statements: assignments, in which the value of an object or the result of a function call is assigned to another object; and automatic print statements, in which values or results not explicitly assigned to an object are printed by an implicit call to a print method. An easy way to control the behavior of a program is to define new classes of data and associated print functions.

In this paper we introduce three new classes of data, display, file, and expr. The display class is used to change the destination of a print statement. Under normal circumstances, printed information goes to the standard output. Objects with class="display" are printed to the "display" device, an alternative destination defined either in an environmental variable options()$display or as an argument to an explicit call to the print.display() function. Typical display destinations are text editors in independent windows on a display screen, printers, or pipes into other software programs. A family of functions display.* has been defined to be sensitive to the environmental variable specifying the display destination.

The mechanics of the display class lead to the second new class of objects. Objects of class display are printed to a system file (using the sink function) in the underlying operating system (usually Unix) and the name of the file is returned as an object of class file. Objects of class="file" are printed by locating the operating system file and printing it by a method appropriate to its subclass. There are various subclasses of the file class: latex files contain input to the LaTeX text processing software (Lamport, 1986), dvi files (device-independent) contain the results of processing by LaTeX, ps files contain information in the page description language PostScript, source files contain the ascii definitions of S objects. The default print method for a file object is to display the ascii text of the file on a display device. The print methods for latex, dvi, and ps use the unix() function to call operating system commands for each of these file types. The print method for source uses the S command source to bring revised function definitions into the working directory.

The third new class, expr, holds statements in the S language of mode expression. By making expressions a class, not just a mode, we can harness the automatic print facilities to aid in debugging. Since we are changing the behavior of the print method for many objects, we must use the default print method to find out what objects we have constructed. By defining the object L and the print method print.expr:

```
> L ← expression(print.default(.Last.value))
> class(L) ← "expr"
>
> print.expr ← function(x) eval(x)
```

typing

```
> L
```

has the same effect as typing the longer statement

```
> print.default(.Last.value)
```

The print methods are implemented as a family of functions print.*. The print.display method is implemented by a display.* family of functions. Examples of both the automatic and explicit use of print.display are in Section 2. We describe the design of the functions in Section 3.

The mechanism leads to a very general interface with other software systems. described and illustrated in Section 4. The statement print.display(x, display="latex") described in Section 4.3 converts the S object x to a file that can be input to the LaTeX typesetting system. The statement print.display(x, display="sas") described in Section 4.5 creates a SAS data set (SAS Institute Inc., 1990).

2. Display of Objects

It is often helpful to have an image of the data visible in a separate window from the one in which the S session is itself running. We provide a set of functions to display text images in their own windows. We assume we are working with an X-window workstation abd tell S which display device to use with the options(display="xedit") statement at the beginning of the S session:

```
> options(display="xedit") #X using xedit
```

We know that we will be working with the S object my.data.frame and decide to assign it the "display" class:

```
> class(my.data.frame) ← c("display", class(my.data.frame))
```

Note that we have placed the new class first, so the automatic print mechanism will find it first, and retained all previous classes. We can now print the data.frame to the display just by typing its name:

```
> my.data.frame
```

The automatic print mechanism recognizes this to be an object of class display and sends it to the print method print.display. Subscripting retains the class of the object.

3. Function Design

The function print.display has been designed as a print method for objects of class=="display". Any object for which class(object)[1] == "display" is automatically printed with the print.display function. Any other S object can be forced to print on the display by explicitly using print.display. The function takes additional arguments of two types. First, it takes general arguments (width= and length=) to prevent folding of long lines, and otherwise take advantage of scroll bars in the displayed window. Second, it takes device-specific arguments that allow user control of fonts and/or pagination on the display device (X.flags=, lpr.flags=, lp.flags=, pr.flags=). We have provided specific functions in the display.* family of function names for 18 different display devices (window systems, screen editors, typesetters, printers, software systems). It is easy for a user with a different software preference or hardware availability to add another similar function.

The visible effect of the print.display function is the appearance of an ascii image of the object on a display device. The mechanism by which this happens is important. We construct an intermediate file, using the S sink function, and forward that file to the print.ascii function, the default print method for objects of class file. We optionally return the name of the file in the "file" attribute of the "result" attribute of the print.display function. The print.ascii function prints the file on whatever device has been defined. Display devices can be defined with options(display="xedit") or by an explicit argument to the print.display function.

Several of our functions create intermediate files in other formats, for example, display.latex creates dvi files (class=c("dvi","file")), which are in turn automatically printed by print.dvi. We have therefore provided several related functions to allow direct manipulation of dvi, PostScript, and scanned image files. We give applications using these display techniques and discuss the construction of the sets of functions designed to work with them.

3.1 print.display as a Method for the print Function

The initial impetus was the Vars function (Harrell 1992a), which collected supplementary information about data.frames (class, factor levels, formats, variable labels) and displayed it in a window. The motivation for the present paper was the recognition that Vars() was a combination of two separable functions. First, it queried a data.frame and constructed a summary of the supplementary information. Second, it displayed its results in a window on the display screen. In this paper, we focus on the design of the display technology and use the initial application as an example.

The user-level function print.display is the print method for objects of class display. It acts like an ordinary method, in that its behavior depends on the class of its argument. It differs from an ordinary method due to its dependence on the normally hidden generic print.ascii function. print.ascii is aware of its environment, more specifically of the value of several components of the options() vector, principally of options()$display. When options()$display is NULL the visible behavior of print.display is identical to that of print: the output is sent to the standard output connected to the S process. When options()$display is non-NULL, the print.ascii function sends the output file created by print.display to the appropriate display device.

The display construct generalizes to include printers and software interfaces. We provide definitions of display="lpr", display="lp" for Unix printer spools. In Section 4.3 we describe the function display.latex to convert an S data.frame to a LaTeX tabular environment. In Section 4.5 we describe the function display.sas to convert an S data.frame to a SAS data file. In both examples, we have options that allow the target programming system to be executed under S control.

3.2 Family of display.* Functions and the Programming Environment

The new generic function print.ascii queries the value of options()$display to determine the specific display device, say it finds "X", and then forwards the temporary file constructed by print.display and any additional arguments to the function display.X. The function display.X uses the arguments and any additional options and then constructs and executes a Unix command for the display. End users will need to set the option(display="X"), but will otherwise not generally work directly with the display.* functions.

The display.* functions are similar in behavior to the generic functions and associated methods of S, in that the user calls the generic print.ascii and lets it decide how it should behave. The difference is that the print.ascii function depends on environmental information, specifically, the value of components of the options() vector, to determine its behavior. The S generic functions depend on only the class of their argument.

3.3 print.file as a Method for the print Function

The function print.file, a method of the generic print, is itself a generic function with methods print.ascii, print.latex, etc. The function print.ascii, the default method for objects of class "file", depends on the environmental information of the options()$display.

Objects in class="file" consist of vectors of file names. A typical example is the vector of names of files resulting from a call such as tmp2 from the function call:

```
> tmp2←print.latex("z.tex",dvi.command="dvips",safe=F)
> # Side effect: the file "/tmp/z3906.ps"is printed using
> # the method defined in options()$ps.command
```

In this example, the print.latex function took the Unix file name z.tex containing a LaTeX fragment, appended the missing statements to created an expanded LaTeX file, typeset the expanded file to create a dvi file, sent the dvi file to dvips for conversion to PostScript, and printed the PostScript file using the method defined in options()$ps.command.

4. Applications

4.1 Information About a Data.Frame

The motivating application Vars is the display of summary information about a data.frame:

```
> my.data.frame ←
+  data.frame(x=1:2, y=factor(c("a","b")),
+  q=structure(3:4, label="Z"))
> my.data.frame
    x  y  q
 1  1  a  3
 2  2  b  4
> Vars(my.data.frame) # default: sort by variable name
     Label   Class   Levels
 q      Z
 x
 y           factor     a b
```

The function Vars returns an object of class "display", therefore the result of the Vars function goes directly to the display (in this example, the terminal).

4.2 Contents of a data.frame

We often wish to view the contents of a data.frame while constructing or interpreting an analysis. Say we have a data.frame with 26 variables, and we are currently studying a model based on columns 11:15. The statement

```
> options(display="jot") # editor with SGI
> print.display(cars93[,c(11:15,1:10,16:26)],
+   X.flags="-font Courier10", width=280)
```

displays the reordered columns. In addition, the width has been increased so each row of the data.frame appears on one row of the output file. The small font allows more columns on the screen simultaneously. The editor's scroll bars are used to move around in the window.

4.3 Display S Objects in LaTeX Documents

An S data.frame is often displayed in LaTeX table and/or tabular environments. We provide the function display.latex, a member of the display.* family, to perform the conversion. At user level, the commands are:

```
> print.display(my.object,
+   display="latex", dvi.command="xdvi")
```

The function display.latex uses the generic function latex. latex converts data.frame and matrix objects using the specific function latex.default, an enhancement of the latex.table package (Harrell 1992b, 1992c). Function objects are converted by latex.function using either the standard LaTeX verbatim environment or the S Example environment (Chambers and Hastie 1993). Lists are converted by latex.list, a recursive function that calls the generic function for each element of the list.

The display.latex function is more complex than the other members of the display.* family. The others, so far, have been essentially alternate output destinations for the mono-width ascii font produced by the generic print function. The LaTeX program uses the complete data.frame structure of the actual S object. It finds the S object in the frame of the function that called display.latex (by backing up through the calling sequence using the sys.parent function) and sends that object to the generic latex function. Users may wish to call latex directly.

The latex.default function uses format.df, a stand-alone function based on Harrell's latex.table package. Numeric, factor, and character data (including imbedded blanks) are correctly formatted. Matrix components of a data.frame are recognized. The function name format.df indicates that the function is a model for a method designed for data.frame objects.

The primary result of the display.latex function is a LaTeX input file fragment.tex that will be pasted into a complete document. Secondary results are the execution of the latex program on the fragment, and display of the result with the print.dvi method. print.dvi uses another family of functions, with generic function dvi and specific functions dvi.*, for the display of the file.dvi files constructed by the latex function. Four examples are provided, dvi.xdvi for X-windows, dvi.dvips for conversion to PostScript, and dvi.lp and dvi.lpr for line printers. When dvi.dvips is used, one of two additional arguments (dvips.command or ps.command) may be used to define a command (ghostview, lp, or lpr, for example) to view the PostScript. When dvips.command is used, dvips output is piped directly to the Unix command, usually a printer spooler. When ps.command is used, dvips creates a PostScript file and then calls print.ps to display the file, usually on a screen viewer that might need to re-read the input file to display an earlier page.

4.4 Display of Related Files

The next example uses the vector of file names of class "file". We have a data.frame constructed by entering data collected by means of a multi-page data collection form. The form is stored on the computer system in a set of files, one per page. There are occasions while studying the data when we wish to see the image of the paper data collection form.

We construct an S variable form.page that records the page number in the form from which each variable was taken. For example,

```
> my.data.frame ← data.frame(x=matrix(1:12,3))
> names(my.data.frame) ←
+   c("id","age","cholesterol","pulse")

> form.page ← paste("Study.R93.124",c(1,2,2,3),sep="/")
> names(form.page) ← names(my.data.frame)
> class(form.page) ← "file"

> print.default(form.page[c(1,4)])
                id                    pulse
 "Study.R93.124/1"    "Study.R93.124/3"
attr(, "class"):
[1] "file"
```

The structure of form.page says that id comes from page 1 of the form and pulse comes from page 3.

Now when we wish to examine a particular page of the form, based on the current view of the data, we can just call it up. We assume the form is stored in ascii text:

```
> form.page["age"]    # print page with "age" question
```

Since form.page is a vector of class "file", the referenced page is automatically printed by the print.file method. This system generalizes to any subclass of file.

4.5 Interface to Programs, SAS as an Example

The print.display function takes an S object and converts it to another format. The examples so far have been conversions to visual formats. Conversion to the system file format of another software system also fits the general definition. Indeed, the *.tex file format is also an example of input to another software system.

In this section we show the conversion to the format of another statistical system. We use SAS (SAS Institute, 1990) as the example. The function display.sas is a member of the display.* family of functions. It is more complex than most members of the family as it must prepare not only a data file but also a stdin file that gives instructions to SAS to reproduce the variable names and the numeric or character values of the variables. It uses the complete data.frame structure of the actual S object by backing up through the calling sequence (using the sys.parent function in the same way as does display.latex). Users will usually choose to call sas directly. Matrix components of the S data.frame are separated into individual columns. Character and factor data are identified as character in the stdin file. Missing values in numeric data and imbedded blanks in character data are converted correctly. sas also uses the format.df function.

The simplest application is moving the data for analysis or display using a technique that is available elsewhere. An extension of the simple application is placing the conversion inside an iteration loop in S and using the S function to drive an iterative technique using procedures available in the other program. A working example of the iteration loop is included with the distribution.

4.6 Use of nroff/troff or LaTeX for S Help Files

The help files in S are written in nroff/troff. When the nroff files are displayed in the S window they often run off the screen and are not visible at the same time as the command for which they provide guidance. We propose three methods to print them in their own window elsewhere on the display screen

The first method is a simple revision of the help function in S. The help.display function places the nroff output on a temporary text file, then sends the name of the temporary file to the display command.

The second method is more complex, but often needed because the nroff program is an option, not automatically distributed with Unix systems. The function help.tex converts the nroff source files from the .Data/.Help directory to LaTeX (using the doc_to_tex files from Chambers and Hastie (1993)), runs the conversion through the LaTeX program, and displays the dvi file on the display screen.

The third method sends the troff output to the screen using a preview.troff program with the sequence

```
IRIS 61% setenv S_LP preview.troff
IRIS 62% S
> help(function.name, offline=T)
```

Availability and Acknowledgments

The programs (Heiberger and Harrell, 1994) are publicly available for download from the the statlib archive at statlib@lib.stat.cmu.edu. This cooperative work was made possible by the availability of the Internet.

References

Becker, Richard A., John M. Chambers, and Allan R. Wilks (1988). *The New S Language: A Programming Environment for Data Analysis and Graphics.* Wadsworth & Brooks/Cole, Pacific Grove, CA.

Carlisle, David P. (1991, 1992). dcolumn.sty, longtable.sty, newarray.sty, here.sty. LaTeX style files available for download from various ftp sites. Originally developed by carlisle@cs.man.ac.uk.

Chambers, John M., and Trevor J. Hastie (1992). *Statistical Models in S.*, Wadsworth & Brooks/Cole, Pacific Grove, CA.

Chambers, John M., and Trevor J. Hastie (1993). "Functions to convert S documentation to LaTeX," programs are available from statlib@lib.stat.cmu.edu. Send e-mail: "send s.to.latex from S".

Harrell, Frank E., Jr. (1992a). "Display of supplementary information from data.frames," programs formerly available from statlib@lib.stat.cmu.edu. Send e-mail: "send vars from S". They have now been superceded by Heiberger and Harrell (1994).

Harrell, Frank E., Jr. (1992b). "S Matrices to LaTeX," programs available from statlib@lib.stat.cmu.edu. Send e-mail: "send latex.table from S". They have now been superceded by Heiberger and Harrell (1994).

Harrell, Frank E., Jr. (1992c). "Translate characters in S character arrays," programs are available from statlib@lib.stat.cmu.edu. Send e-mail: "send translate from S".

Heiberger, Richard M, and Frank E. Harrell, Jr. (1994). "S Functions for Screen Display of S Objects," programs are available from statlib@lib.stat.cmu.edu. Send e-mail: "send print.display from S".

Lamport, Leslie (1986). *LaTeX: A Document Preparation System.* Addison-Wesley, Reading, MA.

Mittelbach, Frank (1990). array.sty. LaTeX style file available for download from various ftp sites. Originally developed by PZF5HZ@DRUEDS2.bitnet.

SAS Institute Inc. (1990). *SAS User's Guide: Basics, Version 4 Edition.* SAS Institute Inc, Cary NC.

Statistical Sciences, Inc. (1991). *S-Plus Reference Manual.* Statistical Sciences, Inc., Seattle.

LISP for Interval Computation

Trong Wu
Department of Computer Science
Southern Illinois University
Edwardsville, IL 62026-1656

Abstract

Most computer users are not aware that placing a numerical value within a computer will result in a different stored value than the one used for input and commit this rounding error without obvious warning. In general, the output of a numerical computation program can only provide an approximated solution and execution of a numerical computation on one machine is different from the output from another. Therefore, numerical computations are often not reliable. However, a method called interval arithmetic may be the solution for detecting the maximum error of the given computation problem. This paper addresses the use of LISP programming techniques to develop reliable components for interval arithmetic. The LISP programs are machine independent and provide self-validated computation.

1. Introduction

All computer systems are finite state machines, they are not able to deal with the computation of real numbers. They can only compute a finite subset of rational numbers. Consequently, any numerical computation using computer systems will involve rounding errors and propogated errors, and the solution is only an approximation to certain problems. A general question should be, "What is the size of error in the result?" In resent years, some mathematicans developed a technique for keeping track of errors. This technique is called interval computation or analysis. These mathematicans considered for each real number x, there is an interval $[a, b]$ such that $x \in [a, b]$ where a and b are real numbers. They treated an interval as a new kind of number. In this treatement, each real number induced two real numbers for computation. In general, both of these two real numbers are not able to be represented correctly in computer systems. The main deficit is that the length of the resulting interval is too big.

To overcome the deficit, we need to create the shortest possible intervals for all initial values for computation. In this paper, we consider for each real number x, two computer floating-point numbers a and b such that $x\in[a, b]$, where b-a is the shortest interval the underlying hardware computer system can provide and the resulting interval will be the smallest we can get. High level programming languages that are commonly used for numerical computation like FORTRAN, the C language, even the C++ language are not able to implement this kind of computation. The author has found that the LISP is a suitable language to implement this interval computation. We have developed some fundamental functions using the Common LISP programming language [6]. The reasons for using the Common LISP are that the programs we developed in LISP are machine independent. LISP does not have a size limitation for integers thus, the language can simulate any floating-point number format with arbitrary number of bits in the *mantissa*, and any complicated computational problem only requires some basic programming skill to implement a complex and difficult algorithm.

A mathematical foundation that supports interval computation is given in section 2. Functions for interval computation is developed in the section 3. Some examples is given in Section 4. Finally, conclusion is followed.

2. Mathematical Foundation

In 1966 mathematican R. Moore [3] proved an important theorem that supports interval computation and since then interval computation has become a new and growing branch of applied mathematics. We will call this theorem the fundamental theorem of interval computation. It is necessary to state the theorem here to support our work.

Theorem Let $f(x_1, x_2, \ldots, x_n)$ be a rational function of n variables. Consider any sequence of arithmetic setps which serve to evaluate f with given arguments $x_1, x_2, \ldots, x_n$. Suppose we replace the arguments x_i by corresponding interval X_i ($i = 1, 2, \ldots, n$) and replace the arithmetic steps in the sequence used to evaluate f by the corresponding interval arithmetic steps. The result will be an interval $f(X_1, X_2, \ldots, X_n)$. This interval contains the value of $f(x_1, x_2, \ldots, x_n)$ for all $x_i \in X_i$ ($i = 1, 2, \ldots, n$).

The proof of this theorem was given by Moore [3]. We will present an example to justify the result of this theorem.

An Example

Consider the function of one variable

$$f(x) = x\,(x^2 - 2).$$

Suppose that we evaluate this function with interval argument X= [−3, 3]. We first compute

$$X^2 = [0, 9],$$

and

$$X^2 - 2 = [-2, 7],$$

then compute

$$X(X^2-2) = [-21, 21].$$

The Theorem guarantees that

$$-21 \le f(x) \le 21 \text{ for all } x \in [-3, 3].$$

The actual range of $f(x)$ for $x \in X$ is [−21, 21]. We have obtained exact bounds on the range of f by an evaluation of f with an interval argument X. Anyhow, f has both a minimum and maximum within interval X.

3. The Interval Arithmetic

The basic argument to use interval arithmetic instead of real numbers is to provide an error bound in solving a numerical computation. In numerical analysis, interval arithmetic will provide an interval that includes all the values in the range of the given mathematical expression over the designated domain for every computation step. The error is controlled within this interval. Some early work in this area are given by Aberth [1], Alefeld and Herzberger[2], Rotschek and Rokne [5] and Moore [3, 4].

Let $A = [a_1, b_1]$, and $B = [a_2, b_2]$ be two closed intervals of real numbers. Then the interval arithmetic operations are generally defined as

(1) Addition

$$A + B = [a_1, b_1] + [a_2, b_2]$$
$$= [a_1 + a_2, b_1 + b_2]$$

(2) Subtraction

$$A - B = [a_1, b_1] - [a_2, b_2]$$
$$= [a_1 - b_2, b_1 - a_2]$$

(3) Multiplication

$$\begin{aligned} A * B &= [a_1, b_1] * [a_2, b_2] \\ &= [\min(a_1 a_2, a_1 b_2, b_1 a_2, b_1 b_2), \\ &\quad \max(a_1 a_2, a_1 b_2, b_1 a_2, b_1 b_2)] \end{aligned}$$

(4) Division

$$\begin{aligned} 1/B &= [1/b_1, 1/a_1], \quad \text{if 0 is not in B} \\ A/B &= A * (1/B) \\ &= [\min(a_1/b_2, a_1/a_2, b_1/b_2, b_1/a_2), \\ &\quad \max(a_1/b_2, a_1/a_2, b_1/b_2, b_1/a_2)] \end{aligned}$$

In a special case, if a real number x is a floating-point number with respect to a given machine then a degenerate interval $[x, x]$ is used for computation.

4. The Implementation

There are two different ways to represent a real number in binary digits, some computer systems use downward rounding and others use upward rounding. In our implementation, we assume that the computer system is using downward rounding. For an inputted real number, the system will provide us an output for the lower bound of the interval. The upper bound of the interval is obtained by adding 1 to the last bit of the lower bound. This method will assure that the obtained interval is the smallest interval for a given real number. The number of digits in the binary representation is arbitrary and is determined by the user at run time.

We have developed a manual driven procedure in LISP to handle interval arithmetic. The procedure includes six functions: addition, subtraction, multiplication, division, conversion real to binary, and conversion binary to real. The precision is arbitrary and determined or entered by user at run time. This can provide the most accurate computation for the given underlying hardware. To execute each function, the user inputs two real numbers, x and y, and the number of desired binary digits. Then the procedure, for each real number, will return an interval that contains the real number in binary digits. Next, the procedure returns the answer of the operation in two forms: a binary interval, and a decimal interval. The format is shown below:

```
Select your function:
addition, subtraction, multiplication, or division
Enter a number: a real number
Enter the number of digits: an integer
The lower bound in binary digits
The upper bound in binary digits

Enter a number: a real number
Enter the number of digits: an integer

The lower bound in binary digits
The upper bound in binary digits

Result =[binary lower bound, binary upper bound]

       =[decimal lower bound, decimal upper bound]
```

5. Sample Results

We have carefully tested the procedure and found that the procedure does predefined functions correctly. The LISP program was run on a MicroVax II machine with the ULTRIX Version 2.0 operating system. Some sample output are given:

1. Addition

```
Enter a number: 3.56
Enter the number of digits: 60

  The lower bound
=11.100011110101110000101000111101011100001010001111010111000010
  01111010111000010
=3.5599999999999999999514277426726494013564661145210266113281250
  5210266113281250

  The upper bound
=11.1000111101011100001010001111010111000010100
  01111010111000011
=3.56000000000000000003816391647148975607706233 8
  5906219482421875

Enter a number: 3.5
Enter the number of digits: 20

  The binary number
= 11.10000000000000000000

  The lower bound
= 11.10000000000000000000
= 3.50000000000000000000

  The upper bound
= 11.10000000000000000000
= 3.50000000000000000000

The result
=[111.000011110101110000101000111101011100001 0
   10001111010111000010,
  111.0000111101011100001010001111010111000010
   10001111010111000011]

=[7.0599999999999999999514277426726494013564661
   145210266113281250,
  7.0600000000000000000381639164714897560770623
   385906219482421875]
```

2. Subtraction

```
Enter a number: 3.56
Enter the number of digits: 60

  The lower bound
=11.1000111101011100001010001111010111000010100
  01111010111000010
=3.5599999999999999999514277426726494013564661 14
  5210266113281250

  The upper bound
=11.1000111101011100001010001111010111000010100
  01111010111000011
=3.5600000000000000000381639164714897560770623 38
  5906219482421875

Enter a number: 2.56
Enter the number of digits: 60
```

```
  The lower bound
=10.1000111101011100001010001111010111000010100
  01111010111000010
=2.5599999999999999999514277426726494013564661145
  210266113281250

  The upper bound
=10.1000111101011100001010001111010111000010100
  01111010111000011
=2.560000000000000000038163916471489756077062338
  5906219482421875

The result
=[0.1111111111111111111111111111111111111111111
   11111111111111111,
  1.0000000000000000000000000000000000000000000
   00000000000000001]

=[0.99999999999999999991326382620115964527940377
   59304046630859375,
  1.00000000000000000008673617379884035472059622
   40695953369140625]
```

3. Multiplication

```
Enter a number: 2.45
Enter the number of digits: 60

  The lower bound
=10.0111001100110011001100110011001100110011001
  10011001100110011
=2.4499999999999999999826527652402319290558807551
  860809326171875

  The upper bound
=10.0111001100110011001100110011001100110011001
  10011001100110100
=2.450000000000000000069388939039072283776476979
  2556762695312500

Enter a number: 2.63
Enter the number of digits: 40

  The lower bound
= 10.1010000101000111101011100001010001111010
= 2.629999999999919964466243982315063476562500

  The upper bound
= 10.1010000101000111101011100001010001111011
= 2.6300000000001091393642127513885498046875
```

```
The result
=[110.01110001100010010011011101001011110001000
  1111111111111111111011110010101100000001000001
  1000100100111000000000000000000000000,
 110.01110001100010010011011101001011110001101
  111001100110011010101001101110100101111000110
  1010011111110000000000000000000000000]

=[6.443499999998039128966745292537293749470232
  79471680992059521020376422484332579188048839
  56909179687500000000000000000000000,
  6.443500000000267393267250337629623815452620
  771557605414614142147478048627817770466208457
  946777343750000000000000000000000]
```

4. Division

```
Enter a number: 3.63
Enter the number of digits: 40

  The lower bound
= 11.1010000101000111101011100001010001111010
= 3.629999999999919964466243982315063476562500

  The upper bound
= 11.1010000101000111101011100001010001111011
= 3.6300000000001091393642127513885498046875

Enter a number: 2.66
Enter the number of digits: 40

  The lower bound
= 10.1010100011110101110000101000111101011100
= 2.6599999999998544808477163314819335937500

  The upper bound
= 10.1010100011110101110000101000111101011101
= 2.66000000000007639755494892597198486328125

The result
=[1.0101110101011010011101110101011010011101000
  1001010100011111010011011011001110011,
 1.0101110101011010011101110101011010011101111
  10110001101110111111001001011101111110]

=[1.3646616541346455174907343363899815669431497
  8221171941186184994876384735107421875,
 1.3646616541354540314929603602400631670414632
  4347846075397683307528495788574218750]
```

6. Conclusion

We have successfully developed a set of functions for interval arithmetic by using the Common LISP language. These functions are machine independent and the precision is determined by the user. The desired number of binary digits is assigned by the user, this number dominates the precision in the computation. The LISP environment stores integers and symbols in a linked list manner of memory cells. It has no range limitation on integer arithmetic; the only limit is its memory size. Therefore, it is ready to use these functions for any applications in interval computations.

References

1. Aberth, O., *Precise Numerical Analysis*, Dubuque: Wm. C. Brown Publisher, 1988.

2. Alefeld, G. and Herzberger, J., *Intrduction to Interval Computations*, Academic Press. New York, 1983.

3. Moore, R. E., *Interval Analysis*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.

4. Moore, R. E., *Methods and Applications of Interval Analysis*, SIAM, Philadelphia, 1979.

5. Ratschek, H. and Rokne, J., *Computer Methods for the Range of Functions*, Ellis Horwood Limited, Chichester, England, 1984.

6. Vax LISP, *VAX LISP/ULTRIX* Version 2.0, *User's Guide*, Digital Equipment Corporation, Maynard, Massachuetts, 1986.

Documentation with Online Programs Rather Than Programs with Online Documentation

John C. Nash, Faculty of Administration, University of Ottawa
136 Jean-Jacques Lussier Private, Ottawa, Ontario, K1N 6N5 Canada
Telephone: (613) 564-6825, Fax: (613) 564-6518, Email: jcnash@acadvm1.uottawa.ca

0. Abstract

Historically, computer programs (and especially those for statistical applications) used separate documentation. Online documentation, especially "help" integrated and customized to the program context, has become a common and useful feature of contemporary programs.

This talk discusses some advantages of operating the program FROM the documentation rather than the documentation from the program. This approach allows users to focus on the task that they want to do, rather than learning the syntax or other details of a feature of the program. That is, we can set aside the peculiarities of the program design and concentrate on what has to be done. Moreover, the documentation can encompass several different pieces of software, allowing comparisons of different programs or their cooperative use for problem solving.

The Software Taxi is a simple hypertext documentation and program launching system developed by the author with Mary M. Nash and assisted by Mary Walker-Smith. The discussion illustrates the idea outlined above and tests its strengths and limitations.

1. The Problem

The growth in the number and size of software packages, particularly those for scientific and statistical computation, presents users and prospective users with a learning-cost problem. That is, it is costly of time and effort, and often money for (unnecessary?) software acquisition, to

- learn about software and its features and style
- learn how to use (statistical) software

This issue software acquisition and use is hardly new. It is, however, exaggerated by the burgeoning size and complexity of packages. Worse, there seem to be few serious attempts to address the issue of software learning costs. This paper considers the learning-cost problem and suggests one way to address it that works with many but unfortunately not all packages.

2. Approaches to Learning about Software

If one can afford the fees and the time, training courses are a very good way to learn about a particular software package. They tend to stress the "how to" aspects of using software rather than its features and capabilities. We must hope that the topics covered are those of interest to us.

If we already have access to a particular software package, then we can read the manual. We should read it again before trying the program. Again, the manuals tend to focus on "how to". If we are in a hurry to do some specific calculation, we may find manuals quite frustratingly detailed. They may presume we have read the material starting at the beginning. The trend to multi-volume manuals renders this approach even less attractive.

"Quick Start" manuals and on-line tutorials could be used to attempt to employ the program quickly. The tools illustrated, such as simple descriptive statistics or regression, may not be the ones we want to learn.

Many of us will simply "try out" a program and look for pointers on its features and method of use. That is, we hope that the **affordances** of the design (Norman, 1992) — the layout, symbols, and other indicators in the user interface — are sufficient to let us infer what can be done and how to do it. This is difficult to arrange in a clear way for mathematical software of any sophistication.

While there has been much investment in "intuitive" user interfaces, it is hardly evident that such developments are helpful to users trying to learn the features and use of statistical software.

- The conventions used by the interface designer may be unfamiliar or not obvious to users.

- Such conventions may not be easy to learn. For example, those who use a strong wrist-hand motion to play piano may move the whole mouse when trying to activate a button.

- The typical icons and simple pull-down menus may be practically useless for even moderately complicated statistical operations. This may be the reason that windowed spreadsheet software has introduced "analysis dialogue boxes" for statistical operations to allow for the setting and adjustment of options and control parameters.

3. User Needs in Software Documentation

The learning issue is intimately connected with that of the design of user-documentation for software. By looking at what the user needs in documentation, we may find some pointers to helpful user-interface features.

First, user documentation needs the actual documentation material to be **well-organized.** We need to know

- what can be done
- how to do it, and
- extra features available.

Beyond the organization of material, users are helped by a good **table of contents** to provide map to the material.

Third, and especially useful to users with some experience, a complete **index** is needed. This should have

- listings of alternate terms for any concepts or topics where different nomenclature or usage exists;
- appropriately distinguished meanings of terms where confusion may arise. For example, chi-squared statistics are used in a variety of situations, so that an unmodified entry under this topic is not very helpful.

Fourth, documentation should include **reasonable examples**. This is an onerous requirement, since the description of non-trivial examples that do not overload the user with a plethora of details requires a good deal of effort and care.

Finally, the documentation must, as concisely as possible, give a **clear description of the software capabilities**.

4. Hypertext

Hypertext provides a good way to structure software documentation. For those unfamiliar with hypertext, the term here refers to a mechanism for displaying units (usually screens) of material in which are embedded menu choices or "buttons" that a user can select to move to another unit or screen of material. The process of moving from one screen to another is called **navigating** the hypertext. In most examples, and in the *Software Taxi* mentioned below, the "buttons" can also start programs.

While hypertext methods offer the potential to present material to users in a convenient and efficient way, we do need to ensure a very good organization of the material to be presented. Details can, and should, be "hidden" in screens that are off the main pathway users are likely to follow when navigating a documentation hypertext.

By running programs under the hypertext manager, we can display graphs and tables, or play sounds or voice messages. Unfortunately, the file formats and file sizes remain a difficulty to the portability of the hypertexts.

Similarly, we can run the program we wish to document, and this is one of the main messages of this paper. That is, after documenting a program feature, we can run the program to illustrate how it works. This gives us our "documentation with online programs", turning around the conventional approach of online "help" within a program. We do not, of course, need to eliminate the latter. Note that this idea offers a twist on the usual "write the documentation, then the program" maxim.

5. The *Software Taxi*

The *Software Taxi* was introduced at the 25th Interface in San Diego (Nash, 1994) as a prototype to test possibilities of running **several** programs to attack a computational problem. The design, partly because of its experimental nature, was of necessity kept deliberately simple and small, yet had to allow easy "jumps" between different hypertext files. Indeed its development was a result of dissatisfaction with two commercial and one shareware systems along with a lack of documentation of such features in some other systems.

The 1993 version of the *Software Taxi* took only a few days of effort to prepare. The latest version, while adding little superficially, has been improved to more easily run other programs, has added utilities to verify hypertext files, allows users to access hypertexts in a given directory directly, and generally is better set up for use by both authors and general users.

We still regard the *Software Taxi* as a prototyping tool to structure and test the material in a hypertext. Since it is based on plain text, the files are portable but are not "fancy". Should the Hypertext Markup Language (HTML) stabilize, we would consider extending the *Software Taxi* with an HTML "front-end".

At the time of writing, the *Software Taxi* is being prepared for distribution. The Level 0 (user) version will be freeware and can be obtained by electronic mail from the author. We hope to install it on various bulletin boards and ftp servers. For those wishing to author hypertexts, there are two other levels of the software incorporating various aids to hypertext preparation and support.

6. Advantages of a Hypertext Approach

The main advantage of the hypertext approach is that we can tell a user about a feature of some software then immediately and directly illustrate what we have just discussed. We need not worry that documentation describes a feature that has been altered or replaced.

More importantly, we only need focus on elements in the software that are of current interest. A large statistical package will have many more capabilities than a single user is likely to be concerned with on a single occasion. Thus, if we are concerned about robust regression, for example, the documentation can discuss the merits of different approaches, the reasons for implementing one or another, the control parameters and other highly specialized matters, and the user can try out and learn about these options without having to learn a great deal of the rest of the package. Moreover, the hypertext scripts provide examples of how to control the program.

Having achieved the title goal of "documentation with online programs", we note that the programs do need scripts. The preparation of these can be assisted by programs that are built into the hypertext. Such **program generators** are a form of tool that could be more generally used to good effect in scientific computation.

In the *Software Taxi* we have found it useful to capture the sequence of screens or actions chosen by a user and to allow automatic playback of such sequences. This is a possible approach for organizing "work in progress", in essence attempting to automate the lab notebook.

7. Disadvantages of the Hypertext Approach

In trying to reduce learning costs, we can remove the bulky manual, but we still require the user to read at least a few of the documentation screens. (You still have to watch the movie, even if you don't wish to read the novel.) Moreover, the hypertexts must be prepared. Even though the *Software Taxi* is designed to make this as simple as possible, it is still a chore.

More seriously, a lot of software cannot be run under control of a script. This is particularly true of such popular tools as spreadsheets. It also applies to almost all software set up for "windowed" operating environments. While there are some intrinsic obstacles to controlling certain graphic operations by scripts (P Velleman, in Goldstein, 1993), it should be relatively simple to provide scripts at the level of "what" to do. However, after nearly a decade, the Apple Macintosh operating environment is just now getting command script capability. (There have been some third-party offerings.)

A final caution with hypertexts is that changes in system configuration (or movement of hypertexts to different systems) can cause unpredictable results. This is, of course, a continuing issue for any program that behaves as an operating shell.

8. Trends

It seems obvious that personal computing equipment such as the Apple Newton and similar book-sized devices are likely to proliferate and become the principal computing interface for many users. Such devices use "pen-based" operating environments and are well-suited to the hypertext / action approach to documentation and use of programs. Moreover, as users need to run more complicated software, graphic icons are more likely to be confusing rather than helpful, especially with a small-screen in uncertain lighting. Plain text allows greater detail to be presented and may be more helpful to users.

9. Summary

In many situations we can document program features and processes then illustrate them "on-line". Moreover, this approach can be simple and effective. However, the need for scripting remains an obstacle to easy porting of the approach to windowed operating environments.

10. References

Goldstein R., (1993) *Statistical Computing: Editor's Notes*, The American Statistician, vol. 47, no. 1, 46-47.

Nash J C (1994) Obstacles to having software packages cooperate on problem solving, Computing Science and Statistics, Volume 25, (Michael E. Tarter and Michael D. Lock, editors), Fairfax VA: Interface Foundation of North America.

Norman, Donald A. (1992) Turn Signals Are the Facial Expressions of Automobiles, Reading, MA: Addison-Wesley.

WHAT IS THE MOST APPROPRIATE SOFTWARE FOR A STATISTICS COURSE?

John D. McKenzie, Jr. and William H. Rybolt
Babson College, Babson Park, MA 02157-0310

Abstract

Last year the authors presented a preliminary report on the advantages and disadvantages of employing a widely-used spreadsheet package in an introductory applied statistics course. In that investigation there was a detailed comparison of the latest versions of Minitab, Microsoft Excel, and Lotus 1-2-3 for classroom use, in which the authors recommended the well-known statistical package over the two popular spreadsheet packages. Since that report, both authors have taught courses with Minitab for Windows and Excel. In addition, both Minitab and Excel have released new versions. In this paper the authors will present an updated recommendation on what is the most appropriate software to use in today's applied statistics courses. This recommendation will be based upon their classroom experiences with Windows-based software and a complete evaluation of the features introduced in the new releases of Minitab and Excel.

Introduction

Babson College is a small private college located in a suburb of Boston. It only offers undergraduate degrees in business and MBAs. All of its students are required to take an applied statistics course or its equivalent. For over 20 years statistical software has been a vital component of these courses. In recent years the platform for such software at Babson has been the school's VAX computer. But with students already exposed to and many businesses moving to a Windows environment, it was determined in the spring of 1993 that this should be the future environment for Babson's statistical software. At that time we began an extensive search for such software. Among the serious possibilities for our 1993-1994 academic year were using statistical software on the VAX for another year, using a popular Windows-based spreadsheet package with statistical capabilities, and using newly developed Windows-based statistical software packages.

At last year's Interface we presented a comparison of Minitab 9.0 for the VAX and two popular spreadsheets, Lotus 1-2-3 and Excel 4.0. Minitab is a very popular package for introductory statistics courses and has been used at Babson since the early 1980s. Among the reasons for considering a change to a spreadsheet package were low incremental cost ($0), a familiar and user-friendly interface (all business students use spreadsheets), and expanded statistical capabilities. After discovering that Lotus, at that time, had only descriptive statistics and regression available, we focused our analysis on Excel 4.0. Initially we were quite impressed by the statistical tools available in Excel 4.0. But soon we became dismayed by some serious problems with Excel 4.0. Thus at Interface '93 we gave a grade of C or D to Excel 4.0, but with the potential of an A grade in the future, and we recommended that users not move to a spreadsheet package at that time.

Shortly after last year's Interface, Minitab announced its first Windows product. After further study Babson decided to use both this product and Minitab on the VAX during the 1993-1994 academic year. We also decided to reconsider our decision in 1994. In this paper we will discuss our latest recommendation for the most appropriate software for our statistics courses.

Windows-Based Software for Statistics

For our 1994 search we did not consider any VAX software due to Babson's migration to Windows. Based upon the school's decision to use Excel and other Microsoft application software throughout the campus, we only examined Excel 5.0. This decision was made even though the latest releases of Lotus 1-2-3 and Quattro Pro have many statistical functions.

We also decided only to examine Minitab for Windows, even though most of the major statistical packages are now released on Windows. Here, with the assistance of a communication from Robin Lock, are some of this software where an asterisk (*) indicates the presence of a student edition: BMDP*, Minitab*, SAS, S-Plus, SPSS*, Stata, Statgraphics, Statistica, and Systat*/Mystat*. We made this decision due to our favorable reaction to Minitab Release 9.0 for Windows and the limited amount of time available to us for a thorough evaluation.

Below is discussion of the similarities and differences between Excel 4.0 and Excel 5.0, and between Minitab 9.0 for the VAX and Minitab 9.0 and 10.0 for Windows. Then there is comparison of Excel, as a representative of the Windows-based spreadsheets with statistical capabilities, and Minitab, as a representative of the Windows-based statistical packages. This is followed by our thoughts about the future.

Excel

Excel is a Windows-based spreadsheet application developed and sold by Microsoft Corporation. Excel is intended to be user friendly with a graphical user interface. Thus it makes extensive use of graphics, the mouse, tool bars, and drop-down menus. It is currently being sold as part of the Microsoft Office Suite and shares Drawing, Equation, WordArt, and other objects with the other applications in the suite. Excel has an installed base running in the millions.

Although the "official list price" of Excel has been around $400. In reality the street price of an Excel upgrade is around $100. The price of an Office upgrade which includes Word 6.0, PowerPoint 4.0, Access 2.0, and Excel 5.0 is under $300. For students, the price of the Office Suite is under $200. Quantity pricing discounts are available for large purchases.

The December 1993 EXCEL.EXE (version 5.0) file has a length of 4,185,600 bytes and the complete Excel Application Package occupies 16,367,256 bytes. Of course, there is some ambiguity associated with these numbers because there are a host of Microsoft Applications such as Drawing, Equation, and WordArt which work with Excel and are not counted in the above numbers.

Due to the fact that Excel is a spreadsheet package and not a statistical package, it is often difficult to find its statistical features. Furthermore, there are only a limited number of statistical capabilities in Excel. For example, it does not handle nonparametric analyses. Excel also has problems analyzing a large data set.

There is only limited external documentation on the statistical capabilities of Excel. For example, it is difficult to determine the computational algorithms used in Excel, although the user guide does suggest that the most appropriate algorithms are not being used. (A discussion on the computational weakness of spreadsheets recently appeared on the Internet Edstat Discussion List.) In addition, Microsoft provides limited statistical support to its users.

The statistical features of Excel are organized into Functions and Data Analysis Tools. The functions are characterized by requiring typically one, two, or three input parameters. These parameters are usually numbers, strings, or ranges. The functions return anything from a single number or string to a complex data structure such as a frequency distribution table in a vertical array.

Examples of functions, along with their Excel descriptions, include

CHIINV(probability, degrees of freedom)
returns the inverse of the chi-squared distribution

COVAR(array1, array2)
returns covariance, the average of the products of paired deviations

FORECAST(x, known y's, known x's)
return a value along a linear trend

NORMDIST(x, mean, standard deviation, cumulative)
returns normal cumulative distribution

Functions are invoked by using the Insert Function or the Function Wizard Button. To understand how these functions are used, consider NORMDIST. The meaning of x, mean, and standard deviation, are relatively obvious. What is not clear is that cumulative is a logical variable: a value of True or 1 causes NORMDIST to return the cumulative value of the normal distribution while a value of False or 0 causes NORMDIST to return the value of the normal density function. To help the user with the choice of inputs, Excel 5.0 has a Function Wizard with labeled boxes for the inputs and a display box for the output. Thus the user can vary the inputs and observe the output before telling the wizard to put the result in the spreadsheet. This feature is new in Excel 5.0 and is an improvement over Excel 4.0 but still needs additional development. The Function Wizard for NORMDIST does not make clear the meaning or possible values of cumulative. Although additional information can be determined by using the online help, this causes a time delay to wait for the help screen to appear. A major improvement would be to display a complete set of information in the Function Wizard Window.

Many of the functions are add-ins which must be brought into Excel before they can be used. Unfortunately before an add-in can be part of a menu it must be loaded even if it is not used during a given session. A better choice would be to include all add-ins on the menu and load them only as needed.

The data analysis tools are invoked by an entirely different mechanism than the functions. To activate a data analysis tool choose Tools from the Command Menu. This is then followed by choosing the Data Analysis subcommand. Since the Data command is next to the Tools command, it is easy to confuse the sequence Tools Data Analysis with Data Analysis Tools. The later sequence does not exist.

After a delay, the InputBox Window appears. This Window features a number of labeled spaces for input. This window look very similar to the Function Wizard InputBox Window. It is very easy to forget whether to invoke a function or to invoke a tool in order to accomplish a particular statistical task.

A more major problem occurs when you actually use one of the tools. Consider the use of the HISTOGRAM tool. While using the HISTOGRAM tool, you discover that a range containing input categories is required in order to construct a histogram. Unfortunately in order to construct such a range you must cancel HISTOGRAM. Then you must construct the input categories without help. If needed you can use the on-line help, but you must remember your exact help topic. Now you must invoke HISTOGRAM again. Obviously if you need to do something in order to execute a command, you should be able to do so by pausing in the middle of the sequence to do whatever is necessary to resume the command sequence. It is hoped that this feature appears in the next version of Excel.

We found a number of calculations which were statistically incorrect. Among these bugs were the naming of a unique mode in bimodal situation., the result of 0 for the maximum when only missing values were present, and the lack of tied rank values. Other serious computational problems involved p-values, output when alpha was specified as zero or one, and regression output from collinear data.

The vocabulary used for describing statistical calculations often represents a poor choice of terminology and surprisingly sometimes is totally inappropriate. Some examples of such errors include the incorrect designation of one-and two-sided p-values, and the specification of a p-value as a test statistic. Another serious terminology gaffe is stating the equivalence of alphas and confidence intervals in performing confidence tests. Examples of additional problems with Excel are available upon request from the authors.

Excel 4.0 versus Excel 5.0

The philosophy behind Excel is to create a basic spreadsheet engine and then enhance it through a variety of specific add-ins. The good news is that this makes it easy to enhance the spreadsheet through a variety of user or commercial macros. This has become especially true since Microsoft switched from the Excel 4.0 macro language to Visual Basic for Applications. This is a user-friendly macro language which makes it easy to write user-friendly, visually-attractive, Windows-based applications. To promote this development Microsoft sells an Excel Developer's Kit, Version 5 for under $50.

The bad news is that this strategy makes macros much slower than if they were compiled to native code. For example, Excel requires 90 to 110 seconds to generate 1000 random numbers with a mean of 10 and a standard deviation of 2. (Not all Excel tasks are slower. To produce a histogram of the above numbers in Excel requires less than 30 seconds.)

In addition to the introduction of Visual Basic, there were many improvements to the user friendliness of Excel with the introduction of version 5.0. Among these were the introduction of sheets, pivot tables, drop down identification labels for tool buttons, and tips. There were also more extensive use of tool bars, improvements to Wizards, and more convenient zoom capability. In contrast, as researched by Derek Upson there were few new statistical capabilities introduced in Excel 5.0. Two exceptions of note were the capability to link graphs and spreadsheets in real time and the capability to specify boundaries for a probability calculation in one of the functions.

When we first examined the statistical features in Excel 4.0 we found problems with the vocabulary used for describing statistical calculations and the accuracy of the calculations. Very few of these problems were fixed in version 5.0. One such correction was the proper computation of the p-value mentioned above. Another correction dealt with the collinear regression output, but in this case another problem was introduced. 0/0 does not equal 65535.

Excel as a Statistical Package

Advantages	**Disadvantages**
Cost	Not a Statistical Package
Large Installed Base	Limited Statistical Support
Known Interface	Slow (Add-In Packages)
Extensive On-Line Help	Inconsistent Design
Visual Basic Macro Language	Lack of Capabilities
Up-To-Date Features Such as Wizards	Computational Concerns
	Poor Choice of Terminology
	Bugs

Minitab

Minitab is a popular statistical package available on a large number of platforms. It is developed and sold by Minitab, Inc. Its mainframe, microcomputer, and PC versions employ an easy-to-use session command interface, while its Macintosh and Windows versions employ graphical user interfaces.

More students have been introduced to statistical software by the use of Minitab than any other piece of software. Examples of its output are contained in a large number of textbooks from a wide range of disciplines. It is also used by analysts in many companies and government agencies.

The academic price for a single copy of Minitab for windows is under $500. Students may purchase the full package for under $200. In addition, a student edition may be purchased for about $50. Quantity purchase prices are also available for academic institutions.

The MINITAB.EXE April 1993 file has a length of 4,227,072 bytes while the Minitab application files total 9,869,097 bytes excluding the data sets.

Minitab contains most of the capabilities needed for standard statistical analyses. In addition, it features a large number of quality tools. There appear to be few difficulties with running large data sets in Minitab.

Minitab provides a variety of well prepared documentation. As with most of the statistical software mentioned above, Minitab uses respected statistical algorithms.

For the most part due to its sole mission of providing statistical tools, Minitab provides a user-friendly environment. Still there are some exceptions: users must type in functions when forming expressions and must leave dialog boxes in order to request help. Due to 20 years of providing statistical software, Minitab presents few problems with terminology and is relatively bug-free.

Minitab 9.0 for the VAX versus Minitab 9.0 and 10.0 for Windows

Minitab 9.0 for the VAX is a powerful member of the Minitab command-driven family. It contains a comprehensive set of statistical capabilities. Among the newer commands in this release are a factor analysis command, a multivariate analysis of variance (MANOVA) command, and many new commands for quality control and design of experiments. With the proper hardware, it can produce a variety of high-resolution graphs. This release of Minitab contains a powerful new macro capability.

Minitab 9.0 for Windows contains all the capabilities of the VAX version (you do not need additional hardware to produce the high-resolution graphs). The major difference between the two versions is the presence of the Windows graphical user interface. Hence Minitab 9.0 for Windows uses a mouse and keyboard to enter commands through drop-down menus, dialog boxes, and even session entries. There are five windows (Data, Session, Info, History, and Graph) in this program. It does not use toolbars or smart keys.

Minitab 10.0 for Windows is the newest Minitab product. It provides additional built-in help along with more powerful data management capabilities. Among these are a direct interface with Excel for the transfer of data and linking data using Dynamic Data Exchange (DDE). New statistical commands in this release are ones for cluster analysis, classical time series analysis, and the design of experiments. There are also many new plots along with the capability to edit and brush graphs.

Excel versus Minitab

We compared the statistical capabilities of Minitab and Excel in ten different areas: descriptive statistics, inference on means, inference on proportions, ANOVA, regression, contingency tables, nonparametric statistics, time series analysis, quality, and probability. In seven cases we concluded that Minitab was clearly superior to Excel. In the case of inference on proportions they were equivalent because neither package performed that analysis. In two other cases, descriptive statistics and probability we rated the packages as equals.

For example, let us consider the descriptive displays (graphs and tables) available in the two packages. A comparison of the graphical capabilities of Minitab for Windows with those of Excel shows that the two packages are roughly equal. Perhaps a slight advantage goes to Minitab in terms of the diversity of graphs produced. A definite advantage goes to Excel in terms of editing and manipulating the graphs which the package produces. We found it extremely awkward and difficult to edit the Minitab graphs in release 9.0, while the Excel editing process soon became almost trivial through the use of the mouse. Minitab 10.0 has greatly improved the easy of editing graphs and is more similar to Excel 5.0.

Minitab does not do three-dimensional scatter, radar, or donut graphs. Excel does not do three-dimensional scatter, control charts, cause and effect, dot charts, unnotched box, or notched box graphs. Both packages did standard bar, grouped bar, stacked bar, histogram, two-dimensional scatter, Pareto diagrams, polygons, high low close, projection, and contour/surface graphs.

Both packages were similar in their ability to generate a variety of tables including cross tabulation, summary, cumulative distribution, frequency distribution, and percentage distribution tables. The pivot table concept in Excel 5.0 makes the manipulation of tables easy with the use of the mouse.

For some tasks, Excel is far slower than Minitab. For example, the random number calculation presented above, requires less than five seconds in Minitab 9.0 for Windows.

Excel has the advantage of a large installed base running into the millions. As a result many users are already familiar with the user interface. There is extensive on-line documentation, but accessing it can be slow on older machines. The addition of Visual Basic makes it much easier to write user-friendly macros. The cost of the statistical features of Excel is almost zero. It is not necessary to buy and support a separate statistical package.

Still Excel is not a true statistical package, even though it provides many essential building blocks. Microsoft only provides limited statistical support. There are problems with trying to analyze large scale problems. Speed is also a problem for Excel. The access to statistical features is somewhat inconsistent. Its statistical documentation is limited. Some of its terminology is poorly chosen. There are concerns about some of its algorithms. Finally, there are far too many computational errors.

Thus we gave a grade of C to Excel 5.0, again with the potential of an A grade in the future. In addition, we recommended the continued use of Minitab 10.0 for Windows for the 1994-1995 academic year.

The Future

In 1993 we were shocked by what we discovered about the statistical capabilities of Excel. How could Microsoft bring a product with so many weaknesses to market? Then we realized that this may be part of Microsoft's corporate strategy. It took the software leader many years to perfect Windows. The first release of Access, one of its data base management programs, also contained many bugs. Perhaps Microsoft is using all of its users as part of gigantic beta test.

Still we were surprised that more of the Excel 4.0 problems were not corrected in Excel 5.0. Going into our 1994 comparison we expected more from Microsoft. Hopefully future releases of Excel will contain fewer bugs, be better documented, and provide easier access to more statistical analyses. Without a doubt more people will analyze data using Excel and other spreadsheets in the future due to their immense base of users and their companies' aggressive pricing policies. In addition, there soon will be well designed add-ins by commercial vendors to enhance the statistical capabilities of spreadsheets such as Excel.

Statistical software vendors should be aware of these strong competitors for their market. In an environment in which only change is constant, they must continue to introduce easier-to-use products with increased capabilities at a low price more frequently. They should also be aware that many purchasers are sadly more concerned with the cost of a product than the accuracy of its algorithms. Hence these vendors must actively consider loss leaders such students editions in order to maintain, or hopefully increase, the number of users who buy their products. Otherwise, they may find themselves with far fewer customers.

Finally, what does the future hold for us users? Still more change. In this market we believe that there will be many new interesting products for us to consider in the future. At Babson we are already preparing for next year's evaluation.

Using Multiple Processors to Compute Robust Regression Estimators

Arnold J. Stromberg and Samuel J. Gardner*

Abstract

Robust regression estimators are notoriously hard to compute. Often algorithms require that fairly simple computations be done on many subsets of the data. Parallel processing machines would be ideal for such computation but they are often not readily available to researchers, and even if available, they often require extensive modification to the code. A far simpler approach is to distribute the computation across several processors on a network. The code is modified to do the computations on a specified portion of the subsets, then the problem is split into pieces and each available processor on the network is used to do a portion of the computation. The results from each processor are then collected and the final answer is computed.

1 Introduction

This paper is presents an improved version of the code used for distributing the computation of the exact least median of squares in multiple linear regression. The new version (available from the first author by e-mail to astro11@ukcc.uky.edu) is shown to be faster than the code discussed in Hawkins, Simonoff, and Stromberg (1994).

2 Distributed versus Parallel Computing

The distinction between parallel and distributed computing is often nebulous but still extremely important. In parallel computing, multiple processors share memory and exchange information while performing a computational task. In an ideally parallelized computation using k processors, the computation would be completed in one k^{th} the time or perhaps even less time. In a distributed computation using k processors, each processor works on a portion of the total computation but the processors do not share memory or other information. The partial solutions from each processor are collected and the final solution is reported. Because the processors do not share information, distributed computations are likely to require more computing time. This is likely to be the reason they have received far less attention in the literature. The major disadvantages of parallel processing are that expensive parallel processing machines are required and that software code is usually machine specific so the user must learn the language for the available machine and then suffer with the fact that the code is not likely to be portable to other parallel processing machines. Distributed processes do not have these disadvantages. They can run on an existing network of CPUs, and the code transfers with at most minor modifications (frequently none!). We believe that these advantages more than make up for the fact that distributed processes may be slightly slower than idealized parallel processing.

3 Steps Required to Distribute a Computation

Hawkins, Simonoff and Stromberg (1994) discuss the steps required to distribute a serial computation across several processors. There steps are:

1. Modify the serial code so that it can compute any portion of the total computation.
2. Identify which processors are available to assist in the computation.
3. Generate input files identifying the part of the computation to be done by each processor.
4. Construct a file that sends the input files and the code from (1) to each processor.
5. Execute the file in (4).
6. Collect the output from each processor and report the final solution.

As an example, they compute the exact value of the least median of squares (Rousseeuw; 1984, Stromberg; 1993) in multiple linear regression. Using these steps and code referenced in Hawkins, et.

*Arnold J. Stromberg is Assistant Professor, Department of Statistics, University of Kentucky, 817 Patterson Office Tower, Lexington, KY 40506. He has support from NSF grant DMS-9204038 and NSA grant MDA-904-92-H-3088. Samuel J. Gardner is a Captain in the U.S. Air Force and is assigned to the Graduate Program, Department of Statistics, University of Kentucky through Air Force Institute of Technology, AFIT/CISP, Wright-Patterson AFB, OH 45433-7765

al., they provide examples showing the effectiveness of this type of distribution. In this paper we will discuss the distribution of the computation of the exact value of the LMS estimate for the data set "educat.dat" found in Rousseeuw and Leroy (1986). Hawkins et. al. report that the median computation time for five runs on one SPARC-IPC was 9705 wall clock seconds (162 minutes). Using four SPARC-IPCs, the median computation time was 2618 seconds (44 minutes). The distributed efficiency is then $2618 * 4/9705 = .95$. This result is quite good, but as Hawkins, et. al. point out, the slowest processor will determine the overall computation time. If one or more of the processors in busy with other jobs, then the computation time could be much longer. For example if one of the processors can only devote 50% effort to the computation then the overall computation time will be close to twice as long.

One solution to the problem of differing loads on the CPUs used in distributing a computation is to split the computation into many small parts and then send the parts one at a time to processors as they become available. In this way, slower processors get fewer of the the parts and the overall computation time is likely to be significantly less than if larger parts were sent to each processor as in Hawkins et. al., thus we suggest the following modification to the steps required to distribute a serial computation:

1. Modify the serial code so that it can compute any portion of the total computation.
2. Identify which processors are available to assist in the computation.
3. Generate a large number of input files splitting the computation into reasonable small parts.
4. Construct a shell script that sends the first *k* input files to *k* available processors.
5. As a processor finishes its computation, the output is appended to an output file for that processor and a new input file is send to that processor.
6. Collect the output from each processor and report the final solution.

Software that implements these steps is available in the software package Chare (Kale, 1990) , used by Raphael Finkel at the University of Kentucky. The disadvantage of Chare is that it is basically a programming language that must be learned and it run only on a very limited number of platforms.

Method	Median (sec)	Perfect[a] Distribution	Distributed[b] Efficiency
SG	2491[c]	2426	97
HSS	2618[d]	2426	93
1 CPU	9705	–	–

[a] median time ÷ 4, 1 CPU

[b] ratio of perfect distribution to median distributed time

[c] $n=12$, $\bar{x}=2525$, $s=65.3$

[d] $n=5$, $\bar{x}=2569$, $s=116$

Table 1: Computation Times for 4 Sparc-IPCs

The Appendix to this paper contains a Bourne shell script that can replace the file "distlms.sh" of Hawkins et. al. The only modifications that need to be made to the other programs provide in Hawkins et. al. are as follows:

1. When prompted by "lmsd.f" for the workstation names, respond with the names of the individual parts of the computation, e.g., p1, p2, ... The shell script requires that the input files have a numbered naming convention. We recommend parts of equal sizes.
2. The program calling.f and its subroutines (which we refer to as "lmsr.f") found in UNIX.PRG (See Hawkins, et. al.) must be compiled for each processor it will be executed on. Modify lmsr.f so that it will print its output to a file called "*host*.out" by adding after "READ(*,10) OUTFIL" the line "OUTFIL = "*host*.out". (*host* is the name given to the machine in the host file for the shell script, e.g. gani.out for the host gani, brahms.out for the host brahms, ...) This must be done for each host.
3. The changes needed to the shell script in the Appendix for the user's network.

As an example, we partitioned the exact computation of the LMS estimate for "educat.dat" discussed above into 50 parts. Table 1 contains the computation times as reported in Hawkins, Simonoff and Stromberg (1994), for their method (HSS) as well as results for the new Stromberg/Gardner (SG) method.

Note that the new method has a better median distributed efficiency. More importantly, note the lower standard deviation of the runs for the new method. The runs for Hawkins, et. al., are more variable because of the fact that the times are highly load dependent, while the new method is less sensitive to varying loads on the network.

As an example of how this new shell script takes advantage of the processors that are not as heavily loaded, the following test case was performed: A numerically intensive program was executed on a Sun Sparc-IPC (host name gani). The distributed computation was performed using four Sparc-IPCs (gani, brahms, bart, and utah). The other three processors were relatively unloaded compared to gani, which could dedicate only 50% of its processing time to the calculations. The total computation time was 3104 seconds, but more interesting was the number of individual parts of the computation that each machine performed: of the 50 total, brahms did 13, utah did 15, bart did 14, and gani did only 8.If each of the machines were equally loaded, then it would be expected that each machine would do 25% of the computation. In this case, gani had a workload that was twice as much as the others, and accordingly it only performed 8/50 = 16% of the computation. Additionally, under the old method (HSS), the total computation time could be expected to be about 5236 (2*2618) seconds because of the higher load on gani. Thus the new method (SG) demonstrates about 60% faster computing time compared to the HSS method.

References

[1] Hawkins, D.M., Simonoff, J.S. and Stromberg, A.J. (1994) "Distributing a Computationally Intensive Estimator: The case of Exact LMS Regression", *Computational Statistics* 9, 83-95.

[2] Kale, L.V. (1990), "The Chare-Kernel parallel programming language and system," *Proceedings of the International Conference on Parallel Processing*, 11-25.

[3] Rousseeuw, P.J. (1984) "Least Median of Squares Regression," *Journal of th e American Statistical Association* 79: 871-880.

[4] Rousseeuw, P.J. and A.M. Leroy (1987) Robust Regression and Outlier Detection, New York: John Wiley and Sons.

[5] Stromberg, A.J. (1993), "Computing the Exact Value of the Least Median of Squares Estimate and Stability Diagnostics in Multiple Linear Regression," *SIAM Journal on Scientific and Statistical Computing*, 14, 1289-1299.

Appendix

```
#!/bin/sh
# Don't delete this first line, it lets the system know which Unix shell
# to use when executing this file.  We are using the Bourne shell, but
# other closely compatible shells could be used with minor modification.

# This is a UNIX shell script that will execute a distributed
# computation of the Least Median of Squares Regression Equation
# (Ref: "Distributing a Computationally Intensive Estimator:  The
# Case of Exact LMS Regression", Computational Statistics (1994), by D.
# Hawkins, J. Simonoff, and A. Stromberg.)  This is a modification of
# the previous method of distributing the simulation, which broke up the
# computation into several parts, one for each machine available, and
# executed the programs remotely.
#
# This method is very much like a multiple server, single line queueing
# system where the shell sends out small pieces of the computation to
# each of the machines, and as these machines become "available" (i.e.
# finish their computation), the shell will send a new job to the
# machine.  The idea is to use the machines which are operating quicker
# more often than the slower ones.
#
# As with all shells, this file must be given execution privilege on
# your machine.  This is done on most machines by the command:
#       chmod +x filename
#
#  Modifications made by Capt Sam Gardner, U.S. Air Force.
#  June 1994

#  Beginning of the Script

HOSTFILE=hostfile
# This variable holds the name of the file which contains the list
# of hosts/processors to use.  Each hostname should be on a separate
# line of the file.

OUTFILE=outfile
# This variable holds the name of the file into which all of the output
# will be put into.

NUMSENT=0
# Variable to count the number of jobs sent.  The input files should
# have a numeric naming scheme, e.g. input1, input.1, input-1, etc...
# In this example (see the rsh command below), the input files are
# named e1, e2,e3, ..., e50

NUMTOTAL=50
# Variable which contains the total number of inputs/jobs to be executed.

#  This checks to see if you created the file defined as HOSTFILE.  If
#  not, the shell informs you and exits.

if [ ! -f $HOSTFILE ]; then
  echo "Cannot find $HOSTFILE"
  echo "Create file $HOSTFILE with a list of hosts to use. Exiting shell"
  exit 1
fi

#  This checks to see if the file defined as OUTFILE above exists.  If so,
#  the shell will ask if you want to delete it.  If you select no, the shell
#  will exit.

if [ -f $OUTFILE ];  then
echo "The output file $OUTFILE exists.  Delete it and continue? (Y or N)"
   read query
   if [ $query = "Y" ]; then
      rm $OUTFILE
   else
     exit 1
   fi
fi

#  Opens an empty file with the name stored in OUTFILE
cat /dev/null > $OUTFILE

# This loop cleans up any files left over from a previous execution of
# this shell and sets up some flag files that the shell needs later on.
# Note that this requires a separate file called "falsefile" which
# contains a single word, FALSE.  Later on a file called "truefile" will
# be needed also, and truefile should contain only the word TRUE.

for name in ‘cat $HOSTFILE‘; do
   rm $name.out 2>/dev/null
   rm $name.sent 2>/dev/null
# puts the word FALSE into the machine.sent file
   cat falsefile > $name.sent
   cat /dev/null > $name.job
done


# The following while loop will check to see first if NUMSENT is less
# than NUMTOTAL.  It then checks to see if a machine/host in the list
# in HOSTFILE is busy.  If not, it sends a remote job to that machine,
# increments NUMSENT by 1, stores the number of the input file, and puts
# the word TRUE in the host.sent file.  If the host is busy, then it
# moves to the next machine.  If a host has completed a job, the file
# host.out will exist and the shell will append the output to OUTFILE,
# remove host.out, and put the word FALSE into host.sent, letting the
# shell know that the host is now available to execute a job.  Otherwise,
# the shell lets you know that the host is busy.

DIRECTORY=directory_where_the_executables_are
```

```
while [ "$NUMSENT" -lt "$NUMTOTAL" ]; do
   for name in `cat $HOSTFILE`; do
      hostflag=`cat $name.sent`
      if [ "$NUMSENT" -lt "$NUMTOTAL" ]; then
         if [ $hostflag = "FALSE" ]; then
            NUMSENT=`expr $NUMSENT + 1`

      rsh $name -n "cd "$DIRECTORY"; $name.lms < e$NUMSENT" &

# Note that the executable files should all have a naming pattern based
# on the host name.  In this case, the executables and input files are
# all in the same directory, with the executable files having names
# "gani.lms", "brahms.lms", etc...  The variable DIRECTORY should be
# changed to the working directory. If the input jobs are names
# differently from e$NUMSENT then that variable should be changed
# accordingly.

            rm $name.sent
            cat truefile > $name.sent
            rm $name.job
            echo "$NUMSENT" > $name.job
            echo "Job $NUMSENT sent to $name"
         elif [ -f $name.out ]; then
            echo "$name completed job, adding $name.out to $OUTFILE"
            rm $name.sent
            cat falsefile > $name.sent
            cat $name.out >> $OUTFILE
            rm $name.out
         else
            echo "job pending at $name"
         fi
      fi
   done
done

#  Now all of the jobs have been sent and the shell will wait until
#  they are all complete.  It will tell you which input file it is
#  waiting on, also, so if you are waiting a long time for one of
#  final jobs to complete, you can kill the shell and run the last
#  job manually on a faster machine.

echo "All jobs have been sent, waiting for final jobs to complete"

for name in `cat $HOSTFILE`; do
   hostflag=`cat $name.sent`
   if [ $hostflag = "FALSE" ]; then
      echo "No jobs pending at host $name"
   else
      job=`cat $name.job`
      until [ -f $name.out ]; do
         echo "Waiting for $name to complete job $job"
```

```
# If this is looping too fast and filling up the screen, the
# following line can be used in the shell  (just delete the pound sign)

#        sleep 5
      done

      echo "$name completed final job, adding $name.out to $OUTFILE"
      cat $name.out >> $OUTFILE
      rm $name.out
   fi

done

# The following line executes the program that computes the LMS fit
# from the output file.

collect

#  Finally the LMS fit is printed to the screen.

more lmsfit

#  End of the shell
```

Multivariate Outlier Detection

David M. Rocke and David L. Woodruff*
Graduate School of Management
University of California, Davis
Davis, CA 95616

Abstract

In this paper we give new insights into why the problem of detecting multivariate outliers can be difficult and why the difficulty increases with the dimension of the data. We then describe significant improvements in methods for detecting outliers and demonstrate using extensive simulation experiments that a hybrid method extends the practical boundaries of outlier detection capabilities. Based on simulation results, we investigate the question of what levels of contamination can be detected by this algorithm as a function of dimension, computation time, sample size, contamination fraction, and distance of the contamination from the main body of data. A more detailed presentation on this topic is contained in Rocke and Woodruff (1994).

1 Introduction

While methods of detection of sporadic outliers in multivariate data have existed for many years (see Hawkins 1980), the problem of detecting clusters of outliers can be extremely difficult. This essentially requires robust estimation of multivariate location and shape, and most estimators for the latter problem are known to fail when the fraction of contamination is greater than $1/(p+1)$, where p is the dimension of the data. Thus detecting outliers or a disparate population that compose more than a small fraction of the data has been impractical in high dimension.

In this paper we give new insights into why the problem of detecting multivariate outliers is so difficult and why the difficulty increases with the dimension of the data. We then describe significant improvements in methods for detecting outliers and demonstrate using extensive simulation experiments that a hybrid method extends the practical boundaries of outlier detection capabilities. Determination of the exact boundaries is complicated by the fact that the probability of detecting outliers depends on many things such as the computer time expended, dimension, number of data points, fraction of data contaminated, type of contamination and algorithm parameters. Nonetheless, based on simulations we are able to specify approximately what levels of contamination can be detected by this algorithm under a variety of conditions.

The estimation of multivariate location and shape is one of the most difficult problems in robust statistics (Campbell 1980, 1982; Davies 1987; Devlin, Gnanadesikan, and Kettenring 1981; Donoho 1982; Hampel et al. 1986; Huber 1981; Lopuhaä 1989 Maronna 1976; Rocke and Woodruff 1993; Rousseeuw 1985; Rousseeuw and Leroy 1987; Stahel 1981; Tyler 1983, 1991). For some statistical procedures, it is relatively straightforward to obtain estimates that are resistant to a reasonable fraction of outliers—for example, one-dimensional location (Andrews et al. 1972) and regression with error-free predictors (Huber 1981). The multivariate location and shape problem is more difficult, since most known methods will break down if the fraction of outliers is larger than $1/(p+1)$, where p is the dimension of the data (Maronna 1976; Donoho 1982; Stahel 1981). This means that, in high dimension, a very small fraction of outliers can result in very bad estimates.

We are particularly interested in obtaining estimates that are *affine equivariant*. A location estimator $\boldsymbol{t}_n \in \mathcal{R}^p$ is affine equivariant if and only if for any vector $\boldsymbol{b} \in \mathcal{R}^p$ and any non-singular $p \times p$ matrix $\boldsymbol{A}$

$$\boldsymbol{t}_n(\boldsymbol{AX}+\boldsymbol{b}) = \boldsymbol{A}\boldsymbol{t}_n(\boldsymbol{X})+\boldsymbol{b}.$$

A shape estimator $\boldsymbol{C}_n \in \mathrm{PDS}(p)$ is affine equivariant if and only if for any vector $\boldsymbol{b} \in \mathcal{R}^p$ and any non-singular $p \times p$ matrix $\boldsymbol{A}$

$$\boldsymbol{C}_n(\boldsymbol{AX}+\boldsymbol{b}) = \boldsymbol{A}\boldsymbol{C}_n(\boldsymbol{X})\boldsymbol{A}^T$$

This implies, for example, that stretching or rotating measurement scales will not change the estimates. Dropping the requirement of affine equivariance does increases

[*] The research reported in this paper was supported by grants from the National Science Foundation (DMS-93.01344) and the National Institute of Environmental Health Sciences, National Institutes of Health (P42 ES04699).

the number of estimators that are available, and there may certainly be cases where a non-affine-equivariant estimator provides superior performance, but it is also important to have robust, computable, affine-equivariant estimators available for use. In fact, though, we know of no non-affine-equivariant estimator that can deal with difficult outliers any better than the best of the affine equivariant methods.

Computational methods have been reported in the literature for a number of approaches for finding robust estimates of multivariate location and shape (and therefore identifying outliers). Combinatorial estimators, such as the minimum volume ellipsoid (MVE) and minimum covariance determinant (MCD) estimators of Rousseeuw (1985; Hampel et al. 1986; Rousseeuw and Leroy 1987), have been addressed with random search (Rousseeuw and Leroy 1987: MINVOL), steepest descent with random restarts (Hawkins 1993a, 1993b: FSA), and heuristic search optimization efforts (Woodruff and Rocke 1993a, 1993b). Iterative estimators such as maximum likelihood and M-estimators (Campbell 1980, 1982; Huber 1981; Kent and Tyler 1991; Lopuhaä 1992; Maronna 1976; Rocke 1992; Tyler 1983, 1988, 1991), and S-estimators (Davies 1987; Hampel et al. 1986; Lopuhaä 1989 Rousseeuw and Leroy 1987) can be computed with a straightforward iteration from a good starting point (Rocke and Woodruff 1993) or using an ad hoc search for the global minimum (Ruppert 1992: SURREAL). Sequential point addition estimators (FORWARD) have been defined algorithmically by Atkinson (1992) and Hadi (1992) working separately. The Hadi paper suggests the use of a non-affine equivariant starting point, but the point addition portion of the algorithm is affine-equivariant and is nearly the same as the point addition portion of Atkinson's completely affine-equivariant algorithm.

In the remainder of the paper, we discuss that nature of multivariate outliers, with a special view to what sorts of outliers are worth studying. We show that outliers that have the same shape as the main data are in some sense the hardest to find, and that the more compact the outliers are, the harder they are to find. We adopt shift outliers as a reasonable target, being of the hardest shape, but of a feasible size to locate. Then we study the comparative performance of the our new hybrid algorithm and previous methods such as MINVOL, FSA, and FORWARD, demonstrating the superiority of the new method. Then we investigate the question of what problems can be practically tackled with our methods.

2 The Nature of Multivariate Outliers

In this section, we investigate the difficulties of locating multivariate outliers. First, to frame the problem as this paper deals with it, we assume that there is a fraction greater than one-half of the data that come from a well-behaved multivariate population, for example multivariate normal. Of course, in practical cases, data transformations may be required before this plausibly holds. In addition to the well-behaved data, there are other data that do not fit the pattern of this well-behaved majority—they may arise from a distinct population, or may be measurement errors; all that is required is that the pattern of these data points is different from the remainder. We will sometimes refer to the majority of the data that come from that well-behaved population as the *good data*, and the remainder as the *bad data*. There is supposed to be no implication that the bad data are necessarily errors—they may just arise from a distinct sub-population—but the locution is convenient.

A second aspect of our viewpoint on this problem is that we aspire to methods that are affine equivariant, so that measurement scale changes or other linear transformations do not alter the behavior of analysis methods. An implication of this viewpoint is that Mahalanobis distances become very important, since these are among the few potentially affine-equivariant outlier identification criteria.

Definition 1 *Let Ω be a positive definite symmetric $p \times p$ matrix. The* Mahalanobis Distance *between points x and y in $\Re^p$ with respect to Ω is defined by*

$$d^2_{\Omega}(x, y) = (x - y)^{\top}\Omega^{-1}(x - y). \tag{1}$$

We refer to the distance and the matrix that defines it interchangeably as a metric.

For data like those we consider here, the *true metric* is the covariance matrix of the population from which the good data arise; a *good metric* is one which is close to the true metric. In particular, when the covariance of the whole sample differs by a lot from the covariance of the good data, a good metric is one that resembles the latter rather than the former.

We will find it convenient to distinguish the size and shape of a metric as follows:

Definition 2 *Let Ω be a matrix defining a metric. The* size *of the metric is the determinant $|\Omega|$. The* shape *of the metric is the equivalence class of metrics Ξ such that $\Omega/|\Omega| = \Xi/|\Xi|$. Equivalently, we may identify the shape as the member of the equivalence class with determinant 1; that is, $\Omega/|\Omega|$.*

This leads to similar definition of shape and size for samples.

Definition 3 *Let $\boldsymbol{X}$ be an $n \times p$ matrix representing a sample of n points in $\Re^p$. Let $\boldsymbol{S} = n^{-1}(\boldsymbol{X}-\overline{\boldsymbol{X}})'(\boldsymbol{X}-\overline{\boldsymbol{X}})$ be the sample covariance matrix. The* size *or* scale *of $\boldsymbol{X}$ is the determinant $|\boldsymbol{S}|$ of its covariance matrix, and the* shape *of $\boldsymbol{X}$ is $\boldsymbol{S}/|\boldsymbol{S}|$. By extension, we refer to the size and shape of other covariance-like estimators, such as the robust ones that are the subject of this paper.*

We now consider the question of what kind of outliers are hard to find. We begin by examining the case in which a good metric is available. This is the goal of most affine-equivariant outlier identification methods—find a good metric so that the outliers will reveal themselves.

Lemma 1 *Consider a sample of n points in $\Re^p$. Let the "good" data have mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$. Let the "bad" data have mean $\boldsymbol{\mu}_0 + \boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Omega}$, and let this comprise a fraction α of the overall data. Then the expected sample mean and covariance matrix are as follows:*

$$E(\overline{\boldsymbol{x}}) = \boldsymbol{\mu}_0 + \alpha\boldsymbol{\mu} \qquad (2)$$

$$E(\boldsymbol{S}) = (1-\alpha)\boldsymbol{\Sigma}_0 + \alpha\boldsymbol{\Omega} + \alpha(1-\alpha)\boldsymbol{\mu}\boldsymbol{\mu}^{\top} \qquad (3)$$

Proof. See Rocke and Woodruff (1994). □

Theorem 1 *Consider a sample of n points in $\Re^p$ Let the "good" data be multivariate normal with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$. Let the "bad" data be multivariate normal with mean $\boldsymbol{\mu}_0 + \boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Omega}$. Consider the Mahalanobis square distance $d^2_{\boldsymbol{\Sigma}_0}(\boldsymbol{x}, \boldsymbol{\mu})$ of a point from the true mean using the true metric. Then, for a fixed location displacement $\boldsymbol{\mu}$ and size $|\boldsymbol{\Omega}|$ of the outliers, the expectation of the Mahalanobis square distance of a bad point from the true mean is least when the shape of $\boldsymbol{\Omega}$ is the same as the shape of $\boldsymbol{\Sigma}_0$. This is thus the worst case from a detection point of view.*

Proof. See Rocke and Woodruff (1994). □

The above theorem implies that the hardest kind of outliers to find, when a good metric is available, is the kind that have a covariance matrix with the same shape as the good data. For this situation, this reduces the infinitely variable kinds of outliers to a single kind. If this kind of outlier can then be detected, so should other kinds. We intend therefore to focus on a situation in which there are good data drawn from a multivariate normal distribution, and bad data drawn from the same distribution and then displaced. These are often called *shift outliers* (Hawkins 1980; Rocke and Woodruff 1993).

Shift outliers may be contrasted with classes of outliers that may be easy to detect, in the sense of appearing disparate even with the bad metric obtained by using all the data. For easily detected outliers, no fancy robust techniques are required, merely examining the Mahalanobis distances from the mean of the data using the covariance matrix of the data will suffice. While we have seen that the shape for bad data that maximizes their masking is the shape of the good data, we have not yet addressed the issue of size. The next theorem shows how easy detection is a consequence of the number and size of the contamination.

Theorem 2 *Consider a sample of n points in $\Re^p$ Let the "good" data be multivariate normal with mean $\boldsymbol{\mu}_0$ and covariance $\boldsymbol{\Sigma}_0$. Let the "bad" data be multivariate normal with mean $\boldsymbol{\mu}_0 + \boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Omega} = \lambda\boldsymbol{\Sigma}_0$, and let this comprise a fraction α of the overall data. Let $\boldsymbol{\Sigma}$ be the expected covariance matrix of the mixed sample as above and consider $d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \alpha\boldsymbol{\mu})$, the Mahalanobis square distance in the bad metric between a data point $\boldsymbol{x}$ and the overall population mean. Then*

1. *The difference in the value of $E(d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \alpha\boldsymbol{\mu}))$ for a bad point and the value for a good point for large η is an increasing function of λ, so that $\lambda = 0$ is the worst case.*

2. *If $\lambda = 0$, so that the outliers form a point mass, and if η is large, then the value of $E(d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \alpha\boldsymbol{\mu}))$ for a bad point is less than the value for a good point whenever $\alpha > 1/(p+1)$.*

3. *If $\lambda = 1$ (pure shift outliers), and if η is large, then the value of $E(d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \alpha\boldsymbol{\mu}))$ for a bad point is always larger than the value for a good point. However, for large p, the distribution of the distance of a good point and the distribution of the distance of a bad point converge.*

4. *For large η, the value of λ at which $E(d^2_{\boldsymbol{\Sigma}}(\boldsymbol{x}, \alpha\boldsymbol{\mu}))$ has the same value for good points and bad points is*

$$\lambda = \frac{(1-\alpha)(\alpha p - (1-\alpha))}{\alpha((1-\alpha)p - \alpha)}$$

whenever this is positive.

Proof. See Rocke and Woodruff (1994). □

Remark 1 *If a good starting estimate for the shape of the good data can be found, then the hardest kind of contamination to discover is that which has the same shape as the good data. Since substantial contamination can only be found by constructing a relatively good shape estimate, this is the most difficult case for such search methods.*

Remark 2 *Although point-mass contamination is the most difficult to detect by the Mahalanobis distance from the sample mean, it is easy to detect in other ways, such as pair-wise distances.*

Remark 3 *Although pure shift outliers might seem to be detectable, given that their mean Mahalanobis distance from the sample mean is larger than that of the good points, no method is known that can find the outliers with complete assurance. This is because the overlap in the distributions is very substantial.*

As we shall see later, pure shift outliers are sufficient to baffle previously proposed methods like the random search algorithm in the program MINVOL (Rousseeuw 1985). Others like those proposed by Hawkins (1992) and Atkinson (1993) turn out to be better than random search. The method proposed in this paper, however, dominates all other methods examined in high dimension.

Because we are mainly interested in high dimension, we will rely primarily on extensive computational experiments to compare methods, rather than the standard, low-dimensional examples often used in the literature. However, we did examine the performance of the code on some of these standard examples, such as the data of Hawkins, Bradu, and Kass (1984), achieving the expected outcomes. For the reasons outlined in this section, the experiments involve mainly pure shift outliers, although a few other cases were examined to check for any sensitivity to this specification. Dimensions as large as 50 were examined, even though the computation times can rise rapidly with the dimension, so that high dimensional cases would be represented. Previously, the literature has concentrated almost exclusively on dimensions less than 10, and usually no larger than five. Methods that appear satisfactory for a problem with three dimensions and 20 data points can be completely impractical for even somewhat larger problems (Woodruff and Rocke 1993a). We examine a range of contamination fractions from $1/(p+1)$, which is the smallest non-trivial amount of contamination, to 40% or 45%, which can be almost impossible to find. There is a theoretical limit on the number of contaminated points that could be found even in principle; the number of good points must be at least $h=(n+p+1)/2$ (Lopuhaä and Rousseeuw 1991). The good data are defined to be multivariate standard normal and the bad data to be multivariate unit normal with a shifted mean. We measure the amount of shift in terms of the unit of measurement $Q_p = \sqrt{\chi^2_{p;0.001}}$, which is more or less the radius of the sphere around the mean that contains almost all the good points. If the outliers are centered at a distance of $2Q_p$, then these spheres should not overlap. We implement outliers at a distance of dQ_p by adding dQ_p^* to each component, where $Q_p^* = \sqrt{\chi^2_{p;0.001}/p}$. This places the outliers at the correct distance out a diagonal. In the experiments used in this paper, we use $d=2$, which we call close outliers, and $d=4$, which we call far outliers.

This generation mechanism is sufficient for use with affine-equivariant methods, but for non-affine-equivariant methods, the data should then be standardized so that the entire sample has mean $\boldsymbol{O}$ and covariance $\boldsymbol{I}$. This can be accomplished using the singular value decomposition as follows. Let $\boldsymbol{S}$ be the covariance matrix of the whole sample of good and bad data. This can be written as $\boldsymbol{S} = \boldsymbol{Q}^{\top}\boldsymbol{D}\boldsymbol{Q}$, where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{D}$ is the diagonal matrix of eigenvalues. If $\boldsymbol{X}$ is the sample, then the sample $\boldsymbol{X}\boldsymbol{Q}^{\top}\boldsymbol{D}^{-1/2}\boldsymbol{Q}$ has the desired properties.

One convenient aspect of the use of shift outliers in this problem is that iterative methods such as M- and S-estimation usually have at most two roots: one that can be found by iterating from the good data (the good root) and one that occurs when iterating from all the data (the bad root). For small amounts of contamination, these may not be distinct, but only when they do differ is the problem interesting.

Finally, we define the criterion of success for an outlier detection method. If the method yields a location $\hat{\boldsymbol{\mu}}$ and a metric $\hat{\boldsymbol{\Sigma}}$, then the method is successful if the largest value of $d_{\hat{\boldsymbol{\Sigma}}}(\boldsymbol{x}, \hat{\boldsymbol{\mu}})$ for a good point is smaller than the smallest value for a bad point. This is a very strict criterion, but some experimentation has suggested that the ordering of the methods is not changed by use of a looser criterion. With pure shift outliers shifted by dQ_p, this is essentially always possible if $d \geq 2$ and if the metric is a good one.

3 Affine-Equivariant Methods for Outlier Detection

All known methods for this problem consist of the following three steps:

1. Estimate a location and metric.
2. Scale the metric so that it agrees on some calibrating distribution.
3. Reject as outliers points whose Mahalanobis distance from the location estimate are sufficiently large.

The last two steps are not difficult, so the essence of the problem comes down to highly resistant estimation of multivariate location and shape. All methods for this problem known to us come down to such an estimation problem. These methods fall into two classes: combinatorial and iterative. Combinatorial estimators construct estimates of location and shape from a subset of the data which itself is hoped to be at least mostly outlier-free. Iterative estimators attempt to satisfy a continuous equation by iteration from a starting point. Unless iteration from the whole sample mean and covariance suffices—an uninteresting case—this requires either direct search or use of a prior combinatorial estimator as a starting point.

Our point of comparison is the random search algorithm MinVol for the MVE (Rousseeuw 1985; Rousseeuw and Leroy 1987; Roussseeuw and van Zomeren 1990, 1991). Until very recently, this was effectively the state of the art.

Our proposed method is outlined below; the rest of this section is devoted to describing the steps in more detail and to comparing the method to those in the previous literature. We will refer to the complete method as the *hybrid algorithm* because it uses both combinatorial and iterative features, as well as incorporating several other useful heuristics.

1. Randomize the order of the data points.
2. Partition the data into $\lfloor n/\gamma(p)\rfloor$ cells indexed by j.
3. For each cell,
   (a) Spend $T/\lfloor n/\gamma(p)\rfloor$ seconds on a Tabu Search for the MCD (Woodruff and Rocke 1993b).
   (b) Use MCD estimate as a starting point for a sequential point addition algorithm using the entire sample of size n starting from the $p+1$ points that have the smallest distance from the MCD location using the MCD metric.
   (c) Use this result as the starting point for translated bi-weight M-estimation (Rocke 1993) using the entire sample of size n. This yields estimates $\hat{\boldsymbol{\mu}}_j$ and $\hat{\boldsymbol{\Sigma}}_j$ of location and shape.
4. Select the index j for which $|\hat{\boldsymbol{\Sigma}}_j|$ is least and set $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}_j$ and $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Sigma}}_j$.
5. Resize $\hat{\boldsymbol{\Sigma}}$ so that the median distance is consistent with an assumed (e.g., normal) distribution; that is, multiply by $\chi^2_{p;h/n}/m$, where m is the hth largest Mahalanobis square distance using the metric $\hat{\boldsymbol{\Sigma}}$.
6. Reject as outliers those points whose Mahalanobis distances exceed a chosen χ^2_p quantile.

3.1 M- and S-Estimation

An S-estimate of multivariate location and shape is defined as that vector $\boldsymbol{t}$ and PDS matrix $\boldsymbol{C}$ which minimizes $|\boldsymbol{C}|$ subject to

$$n^{-1}\sum \rho\left([(\boldsymbol{x}_i-\boldsymbol{t})^\top \boldsymbol{C}^{-1}(\boldsymbol{x}_i-\boldsymbol{t})]^{1/2}\right) = b_0 \qquad (4)$$

which we write as

$$n^{-1}\sum \rho(d_i) = b_0. \qquad (5)$$

It has been shown by Lopuhaä (1989) and, using a different method, by Rocke (1993), that S-estimators are in the class of M-estimators with standardizing constraints with weight functions $v_1(d) = w(d)$, $v_2(d) = pw(d)$, $v_3(d) = v(d)$, where $\psi(d) = \rho'(d)$, $w(d) = \psi(d)/d$, $v(d) = \psi(d)d$, with constraint (5).

In Rocke (1993) it is shown that S-estimators in high dimension can be sensitive to outliers even if the breakdown point is set to be near 50%. We utilize the translated biweight (or t-biweight) M-estimation method defined in Rocke (1993), with a standardization step consisting of equating the median of $\rho(d_i)$ with the median under normality. This is then not an S-estimate, but is instead a constrained M-estimate.

In accord with the theory in Rocke (1993), we have found that the use of the t-biweight M-estimator makes a large improvement in the performance of the hybrid algorithm compared to the use of biweight S-estimation, at least when the outliers lie relatively close in ($d = 2$). When $d = 4$, use of one iterative estimation method or the other made no important difference. Some detailed evidence is given in Table 1. The situation here is that twenty replicates of shift outliers at $d = 2$ and with indicated sample size, fraction of outliers, and computation time allowed (all computation times are CPU seconds on a DECStation 5000/200). The response is the percentage of replicates for which the indicated estimator achieved the good root. Note that the t-biweight performance exceeds that of the biweight S-estimate by large amounts in every case. A large number of additional experiments confirm this important difference in performance.

3.2 Partitioning

The simple iteration scheme for M-estimation fails without a good starting point. An M-estimator that begins iteration using an estimate based on all the data breaks down with $1/(p+1)$ of the data contaminated (Maronna 1976). Two methods of addressing this problem seem possible. One is to look directly for the global minimizer of the S criterion. The other is to find a good starting

Table 1: Comparison of Biweight S-estimation with t-biweight M-estimation. The columns headed "%" are the percentage of 20 trials that the given estimator correctly identified the outliers.

n	α	time (sec)	biweight %	t-biweight %
50	0.30	22	5	50
50	0.30	202	5	70
50	0.35	22	0	20
50	0.35	202	0	25
200	0.30	60	55	95
200	0.30	240	55	95
200	0.35	60	0	35
200	0.35	240	0	55

point for the iteration by use of a preliminary combinatorial estimator.

Ruppert (1992) proposed an algorithm called SURREAL for direct search for the global minimizer of an S estimator used in multiple regression. He reported computational experiments that demonstrated the effectiveness of the SURREAL for this purpose. In the same paper, he proposed an extension of the method to robust estimation of multivariate location and shape. It appears SURREAL is not as effective for this problem as for regression. In dimension 10, SURREAL rarely found the good root when the fraction of contamination was greater than about 12%. Since this was not competitive with other algorithms examined, detailed results are not presented.

We also have examined direct search as a method of finding the good root for S- or M-estimation and have found that it seems superior to use a preliminary combinatorial estimator such as the MCD (Rousseeuw 1985). As pointed out by Woodruff and Rocke (1993b), the use of the MCD to find a good starting point presents severe computational difficulties. Regardless of which algorithms are used to compute them, combinatorial estimators such as the MCD search a space that increases exponentially with the sample size and the dimension. In fact, when using the MCD as a first stage in a two-stage estimator, one can have the perverse situation of being made worse off by having more data. To cope with this problem, the data must be partitioned so that the search space for the MCD is kept in a reasonable range. After some modest experimentation, we settled on a cell size of $\gamma = 5p$. This may possibly be too small for high dimension, but determining the optimal value was beyond the scope of the present paper.

As shown in Woodruff and Rocke (1993b), use of data partitioning in this fashion allows the acquisition of the good root with high probability with a computational time increasing only linearly with n (instead of exponentially).

3.3 Sequential Point Addition

Working separately, Hadi (1992) and Atkinson (1992) have proposed algorithms which begins with an estimate of shape and location based on $(p+1)$ points and then selects successively larger sets—the set with $k+1$ points is consists of those points whose Mahalanobis distances from the mean of the k-set using the covariance of the k-set as a metric are smallest. Because Atkinson's method is completely affine equivariant, we concentrate on this rather than the method suggested by Hadi.

Atkinson's method is affine equivariant. He suggests restarting the procedure many times with randomly selected sets of $p+1$ points. For each trial, sequential addition is performed and for each stage in the sequential addition, the covariance matrix is calculated, and the resulting shape matrix is expanded (or contracted) so that half (or (n+p+1)/2) of the points are included in the ellipsoid defined by the current location and shape. The estimate over all trials and over all stages of each trial in which the scaled shape matrix has minimum determinant may be taken as the robust estimate of the shape and location of the data. Atkinson's algorithm is a large improvement over MINVOL. In the remainder of the paper, we refer to this procedure, following Atkinson, as the forward algorithm, or FORWARD for short.

We found that including a sequential addition step between Tabu search for the MCD and the iterative estimator improved the results in some cases. Here the preliminary MCD estimator is used to choose the $p+1$ points closest to the location estimate using the MCD metric, and then sequential addition as used by Atkinson proceeds once, yielding a new location and shape estimator that is then use to start the iterations for the M- or S-estimator. The importance of including the point addition sub-algorithm is reduced if the contamination is further away from the good data. So, although the inclusion of the point addition sub-algorithm is not critical, it seems well worth the small effort required to code it.

3.4 Minimum Covariance Determinant

Faced with a subsample of contaminated data, our experiments indicate that the best way to find a good starting point for sequential point addition (or for M-iteration) is to search for the MCD. It was originally thought that the MVE would be preferable for computational reasons (see Rousseeuw and Van Zomeren 1990), even though the MCD has greater asymptotic efficiency. This was based on the notion that MVE algorithms would make use of elemental subsets. Woodruff and Rocke (1993a) demonstrated that heuristic search algorithms that use larger subsample sizes perform better. Given this fact, there is no longer any reason to prefer the MVE to the MCD. Simulations done by Woodruff and Rocke (1993b) strongly support the contention that the MCD is in fact the better estimator to use.

The MCD for any set of data is defined by the half sample whose covariance matrix has minimum determinant. It is convenient to search for MCD half-samples moving from half sample to half sample by the removal of one point in the current half sample and the addition of one not currently in. Neighborhoods defined in this way can form the basis of a steepest descent to a local minimum. Hawkins (1993b) suggests the use of steepest descent with random restarts, which he calls FSA. Woodruff and Rocke (1993b) advocate the use of a steepest descent based meta-heuristic called tabu search (Glover 1989, 1990). A tabu search (TS) algorithm for the MCD is given in Rocke and Woodruff (1994).

3.5 A Comparison of Algorithms

Given that some runs in high dimension may take up to an hour of CPU time, and that there are many conditions under which one should compare estimators, a comprehensive Monte Carlo study is impractical. In this section, we compare our algorithm with random search over elemental subsets (Rousseeuw 1985: MinVol). Comparisons with the forward algorithm (Atkinson 1992: Forward), steepest descent with random restarts (Hawkins 1993b: FSA), and Surreal may be found in Rocke and Woodruff (1994). The hybrid algorithm proved to be superior to these methods, as well as to MinVol, for high dimension or large data sets.

The good data in the simulation are multivariate standard normal; the bad data are multivariate normal with covariance $\boldsymbol{I}$ but with a mean displaced a distance of dQ_p, where values $d = 2$ and $d = 4$ were used. The dimension p was 10, 20, and 50, with sample sizes of $n = 5p$, $n = 10p$, and $n = 20p$. Several processing times t were tried for each case, varying from a few seconds to several hours in high dimensional examples. The degree of contamination α was varied from levels where the solution could almost always be found by most methods to levels where none of the methods could get them right.

In order to increase the utility of the number of runs that were practical to perform, a generalized linear model was fit to the outcomes of the experiments, which each consisted of 20 trials at each case. The logit of the probability that a given estimator would succeed in identifying all the outliers was taken to be a linear function of n, α, and $\log(t)$ and their interactions (nonsignificant interactions were removed). Different models were fit for each estimator, distance of outliers, and for each dimension examined.

Table 2: Fitted Performance Measures for the Hybrid Algorithm vs. MinVol in Dimension 10. The columns headed "%" are the predicted percentage of trials that the given estimator correctly identifies the outliers.

α	n	time (sec)	Hybrid %	MinVol %
.1	100	100	100.0	69.2
.1	200	400	100.0	88.9
.2	100	100	99.8	11.0
.2	200	400	100.0	15.9
.3	100	100	83.2	0.7
.3	200	400	97.9	0.4

Table 2 shows the fitted probability of success for some choices of the amount α of contamination, the number of data points, and the estimation time for the hybrid algorithm and MinVol in dimension 10. The clear superiority of the hybrid algorithm is apparent. In higher dimension, limited trials suggest that the the hybrid algorithm is even more dominant. However, given the finite time available for computer simulations in high dimension, most of the runs were devoted to determining the envelope of feasible solution for the hybrid algorithm, rather than to documenting the exact degree of superiority over competing algorithms.

4 Estimating the Envelope

This section is devoted to the following question: for what dimensions, sample sizes, outlier distances, fractions of outliers, and computation times is the hybrid algorithm effective? The theoretical results in Woodruff and Rocke (1993b) demonstrate that any amount of contamination less than 50% can theoretically be handled with sufficient data and sufficient processing time. Here we ask a different question: what amount of contamination can be practically detected with an amount of data that is given and with practical processing times.

Table 3 shows some results. For each indicated combination of dimension and outlier distance, a generalized linear model was fit as described above. Then the level of contamination was found that allowed a predicted 90% of

Table 3: Critical Contamination Level for 90% success with the Hybrid Algorithm. The column headed α is the amount of contamination such that the hybrid algorithm is predicted to be able to identify the outliers correctly in 90% of the instances.

p	d	n	time (sec)	α
10	2	50	200	0.27
10	2	100	200	0.29
10	2	200	200	0.32
10	4	50	200	0.29
10	4	100	200	0.32
10	4	200	200	0.36
20	2	100	800	0.21
20	2	200	800	0.24
20	2	400	800	0.27
20	4	100	800	0.24
20	4	200	800	0.25
20	4	400	800	0.28
50	2	200	5000	0.15
50	2	400	5000	0.16
50	2	800	5000	0.17

the data sets to be successfully completed. To avoid undue extrapolation, computation times and sample sizes were set to within the bounds of what were used for problems of that nature in our study.

The more data (and the more computation time), the greater the fraction of outliers that can be handled. Within our self-imposed bounds, we can say that outlier fractions in the 30–35% range can be reliably solved in dimension 10, with 20–30% in dimension 20 and 15–20% in dimension 50. Although these bounds are crude, it does give some feel for what problems are feasible. It is likely that the sample sizes and processing times for dimension 50 are actually a lot too small. For assured success with high contamination, substantially larger values of both than the ones we used may very well be necessary.

A point that should not be overlooked is that advances in processor technology and parallel processing can have an important effect. For example, a DEC 3000/400 Alpha AXP workstation is about 6 times faster than the DECStation 5000/200 on which these simulations were conducted, and multiple processor machines could also be used to multiply the effectiveness of the algorithm, which is parallelizable in a number of ways (Woodruff and Rocke 1993a).

5 Conclusions

In this paper, we have investigated the nature of multivariate outliers and methods for their detection. We have shown that shift outliers provide a reasonable testbed for multivariate outlier detection, being difficult but not impossible to detect. Using this testbed, we have shown a new hybrid algorithm to be superior to existing methods for this problem. Given sufficient data and processing time, even heavily contaminated data in high dimension can be dealt with.

References

Andrews, D. F., Bickel, P. J., Hampel, F. R., Huber, P. J., Rogers, W. H., and Tukey, J. W. (1972) *Robust Estimates of Location: Survey and Advances*, Princeton: Princeton University Press.

Atkinson, A. C. (1992) "Fast Very Robust Methods for the Detection of Multiple Outliers," manuscript.

Campbell, N. A. (1980) "Robust Procedures in Multivariate Analysis I: Robust Covariance Estimation," *Applied Statistics*, **29**, 231–237.

Campbell, N. A. (1982) "Robust Procedures in Multivariate Analysis II: Robust Canonical Variate Analysis," *Applied Statistics*, **31**, 1–8.

Davies, P. L. (1987) "Asymptotic Behavior of *S*-Estimators of Multivariate Location Parameters and Dispersion Matrices," *Annals of Statistics*, **15**, 1269–1292.

Devlin, S. J., Gnanadesikan, R., and Kettenring, J. R. (1981) "Robust Estimation of Dispersion Matrices and Principal Components," *Journal of the American Statistical Association*, **76**, 354–362.

Donoho, D. L. (1982) "Breakdown Properties of Multivariate Location Estimators," Ph.D. qualifying paper, Department of Statistics, Harvard University.

Glover, F. (1989) "Tabu Search—Part I," *ORSA Journal on Computing*, **1**, 190–206.

Glover, F. (1990) "Tabu Search—Part II," *ORSA Journal on Computing*, **2**, 4–32.

Hadi, A. S. (1992) "Identifying Multiple Outliers in Multivariate Data," *JRSSB*, **54**. 761-771.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986) *Robust Statistics: The Approach Based on Influence Functions*, New York: John Wiley.

Hawkins, D. M. (1980) *The Identification of Outliers,* London: Chapman and Hall.

Hawkins (1993a) "A Feasible Solution Algorithm for the Minimum Volume Ellipsoid Estimator," *Computational Statistics*, in press.

Hawkins (1993b) 'The Feasible Solution Algorithm for the Minimum Covariance Determinant Estimator," Computational Statistics and Data Analysis, in press.

Hawkins, Douglas M., Bradu, Dan, and Kass, Gordon V. (1984) "Location of several outliers in multiple regression data using elemental sets," *Technometrics*, **26**, 197–208.

Huber, P. J. (1981) *Robust Statistics*, New York: John Wiley.

Kent, J. T. and Tyler, D. E. (1991) "Redescending M-estimates of multivariate location and scatter," *Annals of Statistics*, **19**, 2102–2119.

Laguna, M., Barnes, J. W., and Glover, F. (1990) "Scheduling Jobs with Linear Delay Penalties and Sequence Dependant Setup Costs and Times Using Tabu Search", *Applied Intelligence*, in press.

Lopuhaä, H. P. (1989) "On the Relation between S-Estimators and M-Estimators of Multivariate Location and Covariance," *Annals of Statistics*, **17**, 1662-1683.

Lopuhaä, H. P. (1992) "Highly efficient estimators of multivariate location with high breakdown point," *Annals of Statistics*, **20**, 398–413.

Lopuhaä, H. P. and Rousseeuw, P. J. (1991) "Breakdown Points of Affine Equivariant Estimators of Multivariate Location and Covariance Matrices," *Annals of Statistics*, **19**, 229–248.

Maronna, R. A. (1976) "Robust M-Estimators of Multivariate Location and Scatter," *Annals of Statistics*, **4**, 51–67.

Rocke, D. M. (1993) "Robustness Properties of S-Estimators of Multivariate Location and Shape in High Dimension," Working Paper, Graduate School of Management, University of California at Davis.

Rocke, D. M. and Woodruff, D. L. (1993) "Computation of Robust Estimates of Multivariate Location and Shape," *Statistica Neerlandica*, **47**, 27-42.

Rocke, D. M. and Woodruff, D. L. (1994) "Identification of Outliers in Multivariate Data," Working Paper, Graduate School of Management, University of California at Davis.

Rousseeuw, P. J. (1985) "Multivariate Estimation with High Breakdown Point," in Grossmann, W., Pflug, G., Vincze, I. and Werz, W. *Mathematical Statistics and Applications, Volume B*, Dordrecht: Reidel.

Rousseeuw, P. J. and Leroy, A. M. (1987) *Robust Regression and Outlier Detection,* New York: John Wiley.

Rousseeuw, P. J. and van Zomeren, B. C. (1990) "Unmasking Multivariate Outliers and Leverage Points," *Journal of the American Statistical Association*, **85**, 633–639.

Rousseeuw, P. J. and van Zomeren, B. C. (1991) "Robust Distances: Simulations and Cutoff Values," in Stahel, W. and Weisberg, S. (eds) *Directions in Robust Statistics and Diagnostics Part II,* New York: Springer-Verlag.

Ruppert, D. (1992) "Computing S-Estimators for Regression and Multivariate Location/Dispersion," *Journal of Computational and Graphical Statistics*, **1**, 253–270.

Stahel, W. A. (1981) "Robuste Schätzungen: Infinitesimale Optimalität und Schätzungen von Kovarianzmatrizen," Ph.D. Thesis, ETH Zurich.

Tyler, D. E. (1983) "Robustness and Efficiency Properties of Scatter Matrices," *Biometrika*, **70**, 411–420.

Tyler, D. E. (1988) "Some results on the existence, uniqueness, and computation of the M-estimates of multivariate location and scatter," *SIAM Journal on Scientific and Statistical Computing*, **9**, 354–362.

Tyler, D. E. (1991) "Some Issues in the Robust Estimation of Multivariate Location and Scatter," in Stahel, W. and Weisberg, S. (eds) *Directions in Robust Statistics and Diagnostics Part II,* New York: Springer-Verlag.

Woodruff, D. L., and Rocke D. M. (1993a) "Heuristic Search Algorithms for the Minimum Volume Ellipsoid," *Journal of Computational and Graphical Statistics*, **2**, 69–95.

Woodruff, D. L. and Rocke, D. M. (1993b) "Computable robust estimation of multivariate location and shape in high dimension using compound estimators," *Journal of the American Statistical Association*, in press.

A Composite Model for the Distribution of "Species" and its Use in Monitoring Pattern Recognition Algorithms

David S. Newman
Boeing Computer Services
P.O. Box 24346, MS 7L-22
Seattle, WA 98124-0346
dnewman@espresso.rt.cs.boeing.com

Abstract

There is a vast literature aimed at estimating the number of species in a diverse population and the frequency distribution of individuals across species. Regardless of the model or estimating method chosen, estimates of the "true number" of species typically have a very large variance. A new model presented here separates the observed class distribution into "well-defined" and "residual" classes with different generating distributions.

1 Introduction

The detection of abnormal or "novel" behavior in dynamic mechanical systems is an emerging area of critical importance in the aerospace industry. Sensors of acoustic or mechanical vibrations, optical patterns, etc. can be deployed in critical locations in an aircraft or space vehicle, and the resulting time series or space-time series can be monitored for unusual changes in a variety of ways. This complex data is typically generated at a very high rate, in very large volume or both, and real-time processing is desirable in some of the applications envisioned. Furthermore, the dynamics of the vehicle are too complex for a model based on physical understanding to be practical. In this environment, engineers are seeking to use neural networks to reduce and analyze the data generated by the sensors.

In order to be effective, neural networks generally require data that has been carefully preprocessed (transformed). Postprocessing of the output of the neural network, to evaluate and interpret its "analysis," is also commonly necessary. The composite species distribution model, which is the focus of this article, was motivated by the need to develop a sensitive statistical performance monitor to postprocess the output of an ART1 neural network (Carpenter and Grossberg [2]) used in the "novelty detection" environment described above. However, this model may be used with any pattern recognition algorithm, and in fact with any classification system. The composite model may also prove valuable in ecology and genetics, and language suggestive of these applications is used whenever appropriate.

2 Background: Neural Network Novelty Detection

In the typical novelty detection application, the ART1 network is presented with long sequences of patterns generated from relatively short subsamples (windows) of the time series in question. The choice of window sizes, window overlap, and preprocessing is highly application-dependent. ART1 creates a binary-coded classification of these incoming patterns "on the fly" by creating internal representations (templates) of new patterns as they are presented. New patterns may classify to an existing template if they are close enough according to a Hamming metric which is highly dependent on the preprocessing selected. Previously generated templates can be modified to some degree by subsequent patterns. Patterns which do not match any existing template generate a new template.

When the rate of generation of new templates slows down or stops, the network is "trained" to recognize the normal behavior of the suite of sensors it is monitoring. Abnormal or novel behavior may be indicated by a sudden increase in the generation of new templates, but in some cases a more sensitive indicator may be needed. In many applications, the rate of activation of at least some of the more "popular" templates encoded during learning appears to be relatively stable. The qualitative use of pattern activation data is discussed and illustrated in the novelty detection context in Newman and Caudell [11].

The ART1 network itself will not be discussed further in this article, and the reader is referred to [11] and [2] for the specifics of ART1 and to surveys of the general neural network literature such as Grossberg [6] or Hecht-Nielsen [7] for further information.

3 Motivation: Estimating the Number of Species

The basic data used to monitor the neural network after N patterns have been processed is the vector of activation frequencies (or distribution of observed individuals among species or classes) $n = (n_1, \ldots, n_c)$, where c is the number of classes which have actually been observed (or templates which have been created). It is reasonable to represent the data as a sample from a multinomial distribution with a fixed but unknown number of classes C. But now C is a parameter to be estimated, and standard methods for estimating the class probabilities $p = (p_1, \ldots, p_C)$ and testing hypotheses based on the multinomial assumption with C known *a priori* are no longer adequate.

Bunge and Fitzpatrick [1] review the literature on estimation of the "true" number C of species or classes from a sample of size N. Direct estimation of C is frequently difficult, even when unrealistic simplifying assumptions are made, causing many authors to base estimates of C on estimates of the *coverage* u of the sample. Coverage is the true proportion of the population represented in the sample; formally, $u = \sum_{\{i:n_i>0\}} p_i$, where for notational simplicity it is assumed that the indices of the observed classes $i = 1, \ldots, c$ are consistent with the indexing of p, and $c \leq C$.

Estimates of u				
N	MLE	Good	Starr	NPMLE
100	1.00	1.00 (.014)	.012	.015
1000	1.00	1.00 (0)	.004	.006
7440	.9999	.9999 (.0046)	.0007	.0024

Estimates of C				
c	MLE	Good	Starr	Chao & Lee
5	5 (222)	5	423	5 [0]
29	29 (12868)	29	7217	29 [0]
86	86.01 $(7.5x10^9)$	86.01	12797	86.01 [0]

Table 1: Estimates of coverage u and true number of classes C using several methods, for various values of sample size N and corresponding numbers c of classes observed

Experiments have been conducted with neural network output using several of the multinomial, infinite population methods cited in [1]. Table 1 shows some typical results. Numbers in parentheses under the point estimates are estimates of standard deviations. Chao and Lee [3] gave estimators for the coefficient of variation of their estimator $\hat{C}$; these are shown in brackets.

The maximum likelihood estimate (MLE), Good and Chao estimates of u and C and Good's estimate of the standard deviation of u indicate that there are no new classes to be discovered, while the Starr, non-parametric MLE (NPMLE) and MLE standard deviation of C indicate that only a tiny fraction of the classes have been discovered. The lack of agreement among the methods, and the inability of any of the estimates based on $(N = 100, c = 5)$ or $(N = 1000, c = 29)$ to predict future behavior undermines the credibility of the fixed-but-unknown C multinomial model. It seems essential to regard C as a random variable.

Returning to the review by Bunge and Fitzpatrick [1], methods which approximate the histogram of p by some kind of parametric model appear to give better results in practice than those which treat p as the primary vector of parameters, according to the authors. The results of Keener, Rothman and Starr [9] are especially pertinent in light of the considerations above. Following a number of previous investigators, they place a symmetric Dirichlet distribution

$$Prob(p|\alpha, C) = \frac{(\Gamma(\alpha))^C}{\Gamma(C\alpha)} \prod_{i=1}^{C} p_i^{\alpha-1} \qquad (1)$$

on p, although they still treat the number of classes C as a fixed parameter; their approach is empirical Bayesian. They consider estimation of C with α known, and also with α unknown, the latter being the more realistic assumption. (Keener et al [9] use m instead of C, and A instead of α.) They tested their methods with numerous examples, and found that quite often one obtained estimates in which $\alpha \longrightarrow \infty$ or $C \longrightarrow \infty$ while αC approached a finite limit which will here be called θ in what follows. In particular they show that in the more common case in which $C \longrightarrow \infty$ while $\alpha \longrightarrow 0$ but $\alpha C \longrightarrow \theta$, the limiting distribution of c in this case is given by the Ewens [5] sampling distribution

$$Prob(c|\theta) = \frac{(\theta)^c}{(\theta)_N} |S_N^{(c)}|, \qquad (2)$$

where $(\theta)_N = \theta(\theta - 1)\ldots(\theta - N + 1)$ and $S_N^{(c)}$ is the Stirling number of the first kind. Furthermore, the total number of classes c observed is a sufficient statistic for θ. Further support for the use of Ewens distribution may be found in Table 1: note the behavior of Starr's [12] estimate and the nonparametric maximum likelihood estimate [4], which suggest that large values of C are possible.

Informal statistical analyses, including graphical examination of the template activation pattern, and numerical examination of changes in the "Pareto histogram" of pattern activations as N increases, suggest that a subset of the classes (in particular the most frequently occurring classes) behave like a standard multinomial distribution, while the rate of activation of many templates is rather erratic. While the rate of addition of new templates generally slows down, it does not always cease, and will suddenly increase if some "new" phenomenon occurs in the data.

The population biology analogue of this phenomenon would occur when some species are dominant (not necessarily in numbers, as in the neural network application), while many species are in some kind of competitive equilibrium. In genetics, highly selected alleles may follow one pattern, while neutral alleles (as in Ewens [5]) follow another.

4 The Composite Species Distribution Model

These observations motivated the model presented here, which is a compromise between a "full multinomial" model and a limiting Ewens distribution. It is presented in Bayesian terms, although clearly an empirical Bayes interpretation is possible. The total number of observed classes c is divided conceptually into w "well-defined" and r "residual" classes. As a consequence, the total number of patterns N is partitioned by the choice of w into $N = N_w + N_r$, the number of well-defined and residual patterns, respectively. A Dirichlet prior distribution is imposed on the multinomial probabilities $q = (q_1, \ldots, q_w, q_{w+1})$ associated with the $w + 1$ classes formed by lumping all the residual classes together into a single class, and appending it to the well-defined classes.

The conditional distribution of r given w is then given by the Ewens distribution (2), with r replacing c, and N_r, the number of patterns classified to residual classes, replacing N, in (2). An approximate natural conjugate prior on θ has been adopted:

$$Prob(\theta|r_o, n_o) \propto \frac{\theta^{r_o}}{(\theta)_{n_o}} \qquad (3)$$

The hyperparameters r_o and n_o can be any small numbers $(r_o \leq n_o)$; the details of this prior are overwhelmed by the data in the examples which have been investigated.

One of three "reference" or "informative" prior distributions $Prob(w)$ is imposed on w. Following the example of York and Madigan [13], the parameter w is regarded as defining the class of models of interest. The three priors are Jeffreys, Rissanen and the negative binomial; see [13] for details.

Finally, the choice of w is not treated as simply a matter of parameter estimation, but rather one of model uncertainty ([8], [10]). The "Occam's window" approach of Madigan and Raftery [10] is used to restrict the number of values of w which are considered reasonable alternatives, and a posterior distribution over w is computed on this restricted set. The rationale is that it is important to know whether or not the number of well-defined classes is itself well-defined by the data, or not; the answer is expected to be highly application specific.

References

[1] Bunge, J. and M. Fitzpatrick (1993), "Estimating the Number of Species: A Review." *Jour. Amer. Statist. Assoc.* **88**, 364-373.

[2] Carpenter and Grossberg (1987), "A Massively Parallel Architecture for a Self-organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics and Image Processing* **37**, 54-115.

[3] Chao, A. and S.-M. Lee (1992), "Estimating the Number of Classes via Sample Coverage," *Jour. Amer. Statist. Assoc.* **87**, 210-217.

[4] Clayton, M.K. and E. W. Frees (1987), "Nonparametric Estimation of the Probability of Discovering a New Species," *Jour. Amer. Statist. Assoc.* **82**, 305-311.

[5] Ewens, W.J. (1972), "The Sampling Theory of Selectively Neutral Alleles," *Theoret. Population Biol* **3**, 87-112.

[6] Grossberg, S. (1988), "Nonlinear Neural Networks: Principles, Mechanisms and Architectures," *Neural Networks* **1**, 17-61.

[7] Hecht-Nielsen, R. (1990), *Neurocomputing.* New York: Addison-Wesley Publishing Co., Inc.

[8] Kass, R.E. and A. E. Raftery (1993), "Bayes Factors and Model Uncertainty," University of Washington Technical Report No. 254.

[9] Keener, R., E. Rothman and N. Starr (1987), "Distributions on Partitions," *Annals of Statistics* **15**, 1466-1481.

[10] Madigan, D. and A. E. Raftery (1991), "Model Selection and Accounting for Model Uncertainty in Graphical Models using Occam's Window," University of Washington Technical Report No. 213. To appear in *Jour. Amer. Statist. Assoc.*, 1994.

[11] Newman, D.S. and T.P. Caudell (1993), "An Adaptive Resonance Architecture to Define Normality and Detect Novelties in Time Series and Databases," *Proceedings of the 1993 World Congress on Neural Networks, vol.* **4**, 166-176.

[12] Starr, N. (1979), "Linear Estimation of the Probability of Discovering a New Species," *Annals of Statistics* **7**, 644-652.

[13] York, J. and D. Madigan (1992), "Bayesian Methods for Estimating the Size of a Closed Population," University of Washington Technical Report No. 234.

USING "S" FOR A BAYESIAN ANALYSIS OF CLEAVAGE SITES WHEN THE AMINO SEQUENCE IS KNOWN

David M. Reboussin, Michael E. Miller, Margaret K. James, and Aśok C. Antony*
Bowman Gray School of Medicine, Department of Public Health Sciences, Winston–Salem, NC 27157
*Indiana University, Department of Medicine, Indianapolis, IN 46202.

KEY WORDS: Glycosyl–phosphatidylinositol anchoring; Polypeptide; Protein.

The site at which a nascent polypeptide is cleaved and a glycosyl–phosphatidylinositol (GPI) anchor is attached is an important and difficult issue in biochemical research. Recently, two alternative methods to determine the locus of cleavage (ω) have been proposed. One is based on elegant experimental biochemical and molecular studies and assigns *a priori* probabilities to possible cleavage sites. The other method uses data from the amino acid frequency analysis and the nascent polypeptide amino acid sequence to predict the locus of cleavage using a Chi–square statistic, but ignores prior information from the biochemical studies. In this paper, we propose a Bayesian approach for this inference which synthesizes both methods. This allows probability statements regarding predictions of cleavage sites to be made. S code for such analyses is described and an example illustrating the impact of different levels of prior information is provided.

1. INTRODUCTION

Although glycosyl–phosphatidylinositol (GPI) anchoring is only a recently recognized form of attachment of proteins to cell membranes, over 50 such proteins have been identified. The synthesis of proteins destined to be GPI anchored is a complex process. (Biological systems create, transport and utilize many different types of proteins which can be thought of as strings of amino acids.) The newly translated, or nascent, full length polypeptide is further processed by removal of portions of the amino–terminal domain and, possibly, addition of carbohydrates or lipids. Nascent polypeptides destined for GPI anchoring contain features in their carboxyl–terminal amino acid sequence which are recognized by a specialized enzyme called GPI–transamidase. GPI–transamidase changes the polypeptide in two ways: (1) it endoproteolytically cleaves a hydrophobic amino acid–rich portion of the carboxyl end, and (2) it attaches a lipid–rich preformed GPI anchor. The mature GPI protein is therefore always a truncated version of the nascent polypeptide. These final processing steps give the mature protein its characteristics of attachment to cell membranes and probably play a role in its function.

Perturbation of the normal mechanism of GPI anchoring of important proteins synthesized in hematopoietic cells is the basis for a severe life threatening hematological disease known as paroxysmal nocturnal hemoglobinuria. Thus, for this as well as other GPI anchored proteins, the effects of altering the natural locus of GPI attachment are of particular interest to the researcher, but this locus must be determined first. By the time such research is contemplated, the cDNA deduced amino acid sequence of the nascent polypeptide, as well as the amino acid analysis of the mature GPI anchored protein, are often available. Unfortunately, the locus at which the GPI transamidase cleaves the nascent polypeptide is not obvious from this information alone. Determining the locus of cleavage is labor intensive from the biochemical point of view, often involving several months of painstaking and expensive analysis.

Recently, however, two alternative methods to determine the locus of cleavage have been proposed. Antony and Miller (1994) used data from the amino acid frequency analysis and the nascent polypeptide amino acid sequence to predict the locus of cleavage using a Chi–square statistic. The other method, based on elegant biochemical and molecular studies by Kodukula, Gerber, Amthauer, Brink and Udenfriend (1993), assigns *a priori* probabilities to possible cleavage sites. We propose a Bayesian approach for this inference which synthesizes both methods. The intention of the present research was to combine these two methods and allow for the inclusion of replicate amino acid frequency data if available. This paper describes the S software written to implement this new method.

2. LIKELIHOOD AND PRIOR

During an amino acid frequency analysis of the mature protein, bonds between the amino acids are broken and the relative frequency of acids present is determined. Adding a standard to the sample provides a measure of scale so that the total number of amino acids in each molecule can be estimated (see Table 1). The process is not, however, without error. It can destroy the amino acid tryptophan. It may also fail to distinguish asparagine from aspartic acid, or glutamine from glutamic acid. As a result, the frequencies are not perfectly determined: systematic as well as random errors occur.

Table 1: Results from an amino acid frequency analysis.

Amino Acid	Frequency	Proportion
A	1.9	0.02
D or N	6.3	0.07
F	0.0	0.00
G	9.1	0.10
I	0.1	0.00
K	5.4	0.06
L	1.0	0.01
P	32.8	0.37
Q or E	26.9	0.30
S	1.0	0.01
T	3.8	0.04
W	?	?
V	1.0	0.01
Total	89.1	1.00

When the sequence of the complete protein is known, the amino acid frequency analysis can be used to make inference on the cleavage point. Antony and Miller (1994) described "computerized exoproteolytic cleavage", which computes and recomputes the frequency of amino acids from the known complete sequence after deleting one acid from the carboxyl–terminal end at a time. Inference for the cleavage point can then be done by comparing each theoretical frequency analysis to the observed amino acid analysis using a Chi–square statistic to compare theoretical and observed frequencies. This method enjoys a high degree of success for prediction; however, the Chi–square approach is not as flexible as a likelihood analysis and is not well suited for the inclusion of prior information.

Information from the amino acid analysis can be thought of in two parts: (1) the proportions of each amino acid and (2) the total number of amino acids. We will assume these two are independent. Let K be the number of different amino acids in the nascent polypeptide. Denote the proportions from the amino acid analysis by $p = (p_1, \ldots, p_K)$, and the total number by T. Denote the theoretical frequencies by $n_\omega = (n_{\omega 1}, \ldots, n_{\omega K})$, and the sum of the theoretical frequencies $\sum n_{\omega i} = N_\omega$, where $\omega = 1, \ldots, N$ is a possible cleavage locus. Then $L(\omega; p, T) = L(n_\omega; p) \times L(N_\omega; T)$

For observed proportions, we assume a Dirichlet distribution parameterized as

$$L(n_\omega; p) = B^{-1}(n_\omega + 1) \prod_{i=1}^{K} p_i^{n_{\omega i}}$$

where B is the generalized beta function. This implies $\mathrm{E}\,(p_i) = (n_{\omega i}+1)/(N_\omega + K)$ instead of $\mathrm{E}\,(p_i) = n_{\omega i}/N_\omega$ which shrinks the observed proportions towards $1/K$. For the total, a good model is harder to specify. One choice is

$$L(N_\omega; T) = \frac{1}{\Gamma(T)} N_\omega^{T-1} \exp(-N_\omega)$$

which is a gamma distribution with mean T and variance T; that is, shape parameter T and scale parameter set to 1. If the observed frequencies are all gamma with scale parameter 1, then the proportions are Dirichlet and the total is gamma with scale parameter 1. We do not consider direct modeling of the variability in observed frequencies, since replicate amino acid analyses are not typically available. If such data were available, it would be straightforward to estimate the scale parameter in the gamma distribution.

Kodukula et al. (1993) conducted empirical studies to determine not only which amino acids could serve as cleavage points in GPI proteins, but also which amino acids could be adjacent to cleavage points. They presented a table for the propensity of each amino acid to be at the cleavage site, and one or two amino acids from the cleavage site (towards the carboxyl–terminal). The "probability" of a specific amino acid being the cleavage site is the appropriate product from Table 2. For example, if serine (S) is at a given location, and the two locations towards the carboxyl–terminal are arginine (R) and alanine (A), the "probability" is $1.0 \times 0.5 \times 1.0 = 0.5$. We can compute such products for each site in the nascent polypeptide, and, after standardizing, treat the collection as a prior probability for cleavage site. We denote the prior distribution $\mathrm{pr}(\omega)$. Although this has some weaknesses, it provides a great deal of quantitative information about cleavage sites. Kodukula et al. suggest that this prior alone correctly identifies the cleavage point about 75% of the time.

The posterior probability for the cleavage site occurring at a given point in the sequence is the product of the likelihood and the prior distribution. That is,

$$\mathrm{pr}(\omega | p, T) = L(\omega; p, T) \times \mathrm{pr}(\omega)$$

where $L(\omega; p, T)$ is the product of the Dirichlet and gamma distributions described above. This posterior can be used to make predictions for the cleavage site and attach probability statements to those predictions.

3. IMPLEMENTATION IN S

We first describe how data for this problem are stored in S and define some special functions. Procyclic acidic repetitive protein (parp) in *T. brucei* will be used as a

Table 2: Relative propensity as cleavage site: Table II from Kodukula et al. (1993)

Amino Acid	ω	$\omega+1$	$\omega+2$
A	0.4	1.0	1.0
R	ND	0.5	ND
N	0.8	ND	ND
D	0.4	0.4	0.1
C	0.2	0.3	0
Q	0	0.1	ND
E	0	0.4	0
F	ND	ND	ND
G	0.4	ND	0.7
H	ND	ND	0
I	ND	ND	ND
L	0.1	ND	ND
K	0	ND	ND
M	0	0.3	ND
P	0	0	0
S	1.0	0.6	0.3
T	0	0.3	0.1
W	0	0.1	ND
Y	0	ND	ND
V	0.1	ND	0.1

running example. The known full sequence is stored as a character vector, with the carboxyl–terminal first:

```
parp <- rev(c(
"A","E","G","P","E","D","K","G","L","T","K","G",
"G","K","G","K","G","E","K","G","T","K","V","S",
"A","D","D","T","N","G","T","D","P","D","P","E",
"P","E","P","E","P","E","P","E","P","E","P","E",
"P","E","P","E","P","E","P","E","P","E","P","E",
"P","E","P","E","P","E","P","E","P","E","P","E",
"P","E","P","E","P","E","P","E","P","E","P","E",
"P","E","P","E","P","E","P","E","P","G","A","A",
"T","L","K","S","V","A","L","P","F","A","I","A",
"A","A","A","L","V","A","A","F"))
```

The amino acid analysis is stored as a numeric vector with labels distinguishing the amino acid types:

```
parp.analysis <-
 c(1.9, 6.3, 0.0, 9.1, 0.1, 5.4,
   1.0, 32.8, 26.9, 1.0, 3.8, 1.0)
names(parp.analysis) <-
 c("A","DN","F","G","I","K",
   "L","P","QE","S","T","V")
```

The protein as it appears to the analysis is stored a factor object in S: levels (types of amino acid) with zero frequency appear in output from the S function `table` and "invisible" levels (e.g. tryptophan) can be excluded.

```
parp.alt <- rev(factor(c(
"A","QE","G","P","QE","DN","K","G","L","T","K",
"G","G","K","G","K","G","QE","K","G","T","K",
"V","S","A","DN","DN","T","DN","G","T","DN","P",
"DN","P","QE","P","QE","P","QE","P","QE","P","QE",
"P","QE","P","QE","P","QE","P","QE","P","QE","P",
"QE","P","QE","P","QE","P","QE","P","QE","P","QE",
"P","QE","P","QE","P","QE","P","QE","P","QE","P",
"QE","P","QE","P","QE","P","QE","P","QE","P","QE",
"P","QE","P","QE","P","G","A","A","T","L","K","S",
"V","A","L","P","F","A","I","A","A","A","A","L",
"V","A","A","F"),exclude="W"))
```

Computerized endoproteolytic cleavage is done using the `table` function. For the entire nascent polypeptide, deletion of the first amino acid, and deletion of the first 30 amino acids:

```
> table(parp.alt)
  A DN F G I K L  P QE S T V
 12  6 2 9 1 7 4 33 32 2 5 3
> table(parp.alt[-seq(1)])
  A DN F G I K L  P QE S T V
 12  6 1 9 1 7 4 33 32 2 5 3
> table(parp.alt[-seq(30)])
 A DN F G I K L  P QE S T V
 2  6 0 8 0 6 1 28 29 1 4 1
```

The function `seq(i)` produces the sequence $1, 2, \ldots, i$. Negative indices in the subset operator `[]` delete observations. Thus the function `table(parp.alt[-seq(i)])` produces the theoretical frequency analysis for the protein with cleavage site "i". The entire set of theoretical frequencies can be determined in two lines:

```
> parp.cec <- table(parp.alt)
> for (i in seq(parp))
+ parp.cec <-
   cbind(parp.cec, table(parp.alt[-seq(i)]))
```

The first ten columns of the resulting matrix are:

```
> parp.cec[,seq(10)]
   parp.cec
 A         12 12 11 10 10 10  9  8  7  6
DN          6  6  6  6  6  6  6  6  6  6
 F          2  1  1  1  1  1  1  1  1  1
 G          9  9  9  9  9  9  9  9  9  9
 I          1  1  1  1  1  1  1  1  1  1
 K          7  7  7  7  7  7  7  7  7  7
 L          4  4  4  4  4  3  3  3  3  3
 P         33 33 33 33 33 33 33 33 33 33
QE         32 32 32 32 32 32 32 32 32 32
 S          2  2  2  2  2  2  2  2  2  2
 T          5  5  5  5  5  5  5  5  5  5
 V          3  3  3  3  2  2  2  2  2  2
```

A function to evaluate the Dirichlet density is:

```
> ddirich
function(p, n, ...)
{
# Dirichlet density
     if(length(n) != length(p))
          stop("length(n) must be length(p)")
     if(sum(p, ...) > 1 + std.tolerance()) {
          warning("sum of p > 1")
          0
     }
     else prod(p^n, ...)/gbeta(n+1, ...)
}
```

The Dirichlet part of the likelihood can be computed for a given ω as

```
> parp.proportions <-
   parp.analysis/sum(parp.analysis)
> ddirich(parp.proportions, parp.cec[,21])
[1] 2.803521e+15
```

For the entire protein

```
> parp.dlik <-
  apply(parp.cec,2,ddirich, p=parp.proportions)
```

The gamma part of the likelihood is computed similarly:

```
> args(dgamma)
function(x, shape = stop("no shape arg"))
> dgamma(sum(parp.analysis), seq(parp))
  [1] 1.650072e-39 1.473514e-37 6.579239e-36
> parp.glik <-
   rev(dgamma(sum(parp.analysis), seq(parp)))
```

To utilize the prior from Udenfriend and co-workers at Roche Labs, several functions taking the complete, known sequence as an argument were written. Four increasingly complex versions were implemented. roche1 assigns equal prior probability to all acids in the set (A, N, D, C, G, L, S, V). roche2 assigns prior probability in proportion to the second column in Table 2 (the ω site only). roche3 takes into account the ω and $\omega + 2$ sites, but ignores the $\omega + 1$ site. roche4 implements the full scheme using all three sites. In the last three functions, amino acids listed as "not done" (ND in Table 2) were assigned a value 0.1, and zeros were set to 0.01.

A convenient summary of this analysis is graphical, and we have written a specialized plotting function called plot.summary.

4. EXAMPLES

We present an example with known sequence and cleavage point: parp. Inspection of the likelihood as well as biological considerations restricts the range of interest to no more than the 60 amino acids closest to the carboxyl–terminal. Both the likelihood alone and the posterior using any of the four priors correctly identify the locus of cleavage.

Figure 1: Posterior for parp using the likelihood only.

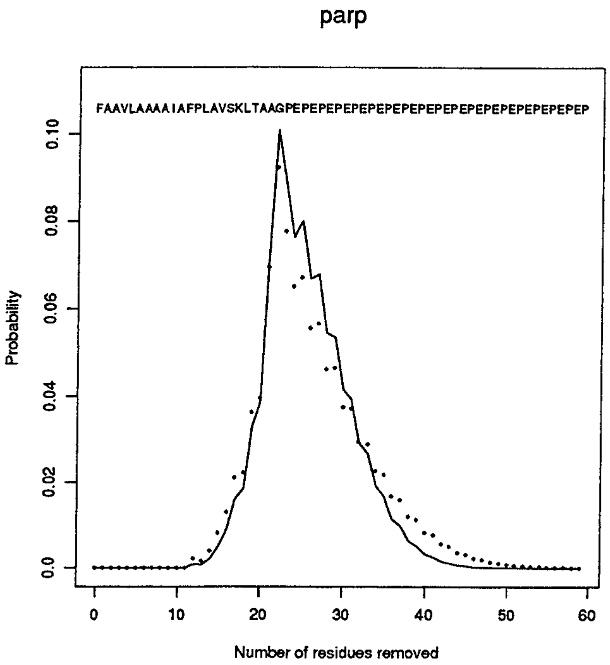


Figure 1 shows the likelihood analysis for parp. The x–axis is the number of amino acids removed by computerized exoproteolytic cleavage, and the y–axis is the posterior probability under a noninformative prior (equal prior probability on all loci). The dots indicate the likelihood based on the proportions from the amino acid frequency analysis, while the solid line indicates the full likelihood. At the top of the graph, symbols for each amino acid in the nascent polypeptide are shown (the carboxyl–terminal amino acid end is to the left). The likelihood has a maximum at the 22nd amino acid, glycine, which is the actual cleavage site for parp.

Figures 2, 3 and 4 represent the impact of adding informative prior information to the likelihood analysis, using roche1, roche2, and roche4. In all three, the posterior concentrates more probability on the correct location, but the posterior (dashed line) based on the full specification suggested by Kodukula et al. (roche4) puts 90% probability on this location.

5. DISCUSSION

The Bayesian approach we propose has a number of

Figure 2: Posterior for parp using the roche1 prior.

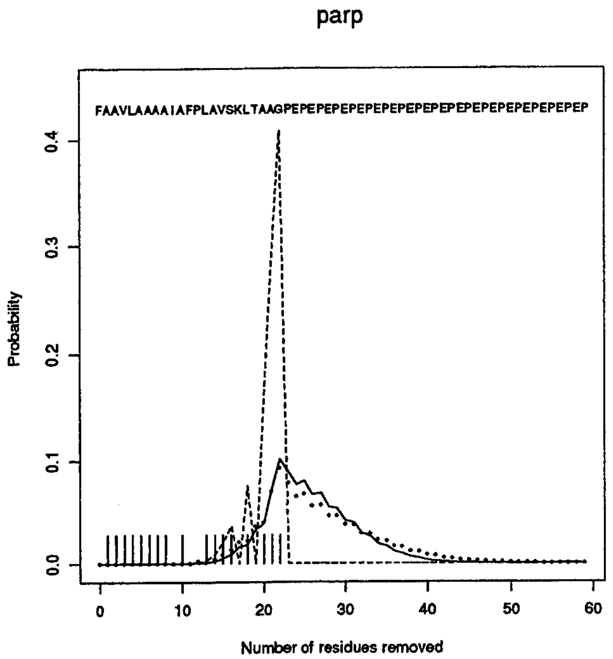


Figure 3: Posterior for parp using the roche2 prior

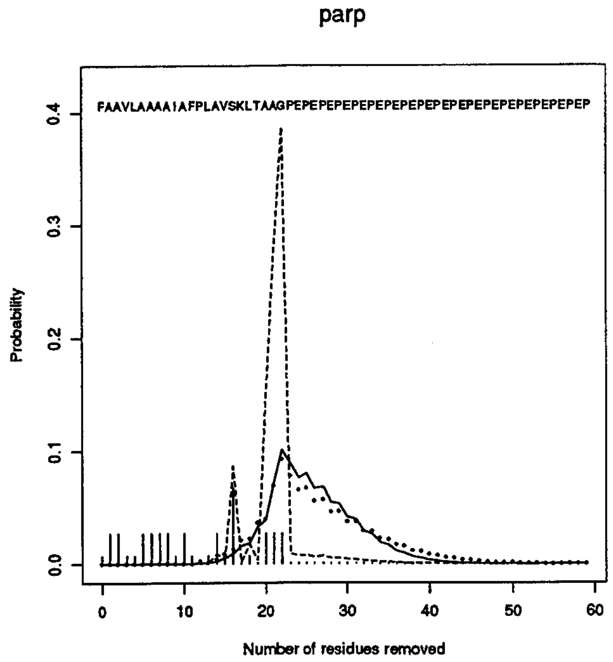
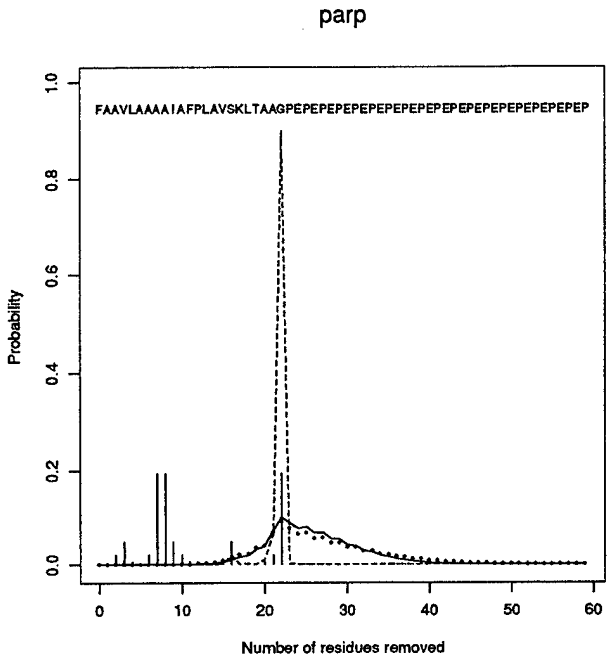


Figure 4: Posterior for parp using the roche4 prior.



advantages over existing methods. It allows both prior information and the amino acid frequency data to contribute to the inference. Replicate frequency data, which may exist though it is not published, can be incorporated directly. By producing a posterior distribution, probabilistic statements concerning most likely cleavage loci can be made: this is important in distinguishing cases where the prediction is highly certain.

The flexibility and presentation quality available from S makes it an ideal computing environment for this problem. The functions written for the Bayesian analysis can be immediately applied to other proteins once the data are input appropriately. Alternative specifications for the likelihood or prior are easily programmed, as are customizations of the graphical summary. Since there appears to be some sensitivity to the prior, rapid recomputation and redisplay is particularly useful.

BIBLIOGRAPHY

Antony, A. C., and Miller, M. E., *Biochemical Journal* 298, pp 9–16, 1994.

Kodukula, K., Gerber, L. D., Amthauer, R., Brink, L., and Udenfriend, S., *Journal of Cell Biology*, 120, pp 657–664, 1993.

Becker, R. A., Chambers, J. M., and Wilks, A. R., *The New S Language*, Wadsworth and Brooks/Cole: Pacific Grove, CA, 1988.

Inference for Lethal Gene Studies via Bayesian Markov Chain Simulation

Jaekyun Lee, Michael A. Newton, Erik V. Nordheim, Hyun Kang*
Department of Statistics, University of Wisconsin-Madison,
1210 W. Dayton Street, Madison, Wisconsin 53706-1685

Abstract

The magnitude of the effect of deleterious genes on a population is classically characterized by the number of lethal equivalents. In conservation and breeding programs, it is often important to be able to distinguish among different combinations of the genetic parameters that lead to the same number of lethal equivalents, for instance, a large number of mildly deleterious genes or a few of fully lethals. This requires, at least, two consecutive generations of mating. Because of the complexity of the likelihood and the existence of many missing data in this two generation case, Bayesian Markov chain simulation is used to infer these parameters and the missing data. In our Markov chain Monte Carlo approach, we introduce a Metropolis-Hastings algorithm for a two dimensional update of parameters having highly attenuated posterior density.

Key Words: Deleterious genes, Lethal equivalents, Bayesian Markov chain simulation, Metropolis-Hastings algorithm.

1 Introduction

The overall effect of deleterious genes on a population is classically characterized by the number of lethal equivalents (Morton, Crow, and Muller, 1956). In a diploid population where deleterious alleles exist at M loci with allele frequency q_i and selection coefficient s_i at the i-th locus, the number of lethal equivalents at the gametic level is expressed as (Morton et al. 1956),

$$\varepsilon = \sum_{i=1}^{M} s_i q_i. \tag{1}$$

At the zygotic level the number of lethal equivalents is $E = 2\varepsilon$. The selection coefficient s_i is defined from the convention that the probabilities of survival of the dominant homozygote, heterozygote, and recessive homozygote are $1, 1 - h_i s_i$, and $1 - s_i$, respectively, and h_i represents the coefficient of dominance at the i-th locus. Lee, Nordheim, and Kang (1994) suggested a new experimental and statistical modeling strategy that leads to an interval estimate of the number of lethal equivalents based on one generation mating design. In conservation biology and breeding programs, it is often important to have more information on the parameters involved in the genetic mortality, i.e., M, q_i, s_i, and h_i, beyond the estimation of the overall mortality effect on a population. To attack this, we introduce a two generation mating model, and construct a corresponding hierarchical likelihood. The complexity of the likelihood leads us apply a Bayesian Markov chain Monte Carlo (MCMC) to enable inference. Data from a two generation mating system ensure identifiability of the selection coefficient. For one generation data, this coefficient is confounded with the number of lethal equivalents (Lee et al., 1994). In our MCMC implementation, a new two dimensional proposal distribution is used for parameter having an attenuated joint posterior distribution.

*North Central Forest Experiment Station and Department of Forestry, University of Wisconsin - Madison

2 Hierarchical Modeling of Two Generation Mating

We consider selfing as the mating system of the experiment here because of its simplicity (see Lee et al., 1994). We assume that the dominance coefficient (h_i) of lethal (=deleterious) alleles is zero, that is, the deleterious alleles are recessive, so that a heterozygous individual with these alleles is always viable. We also assume a single lethal allele frequency q per gamete over all loci carrying lethals (M loci), and single selection coefficient s for these lethal alleles. The selfing experiment has several hierarchical steps. First, we assume that our base (monoecious diploid) population has M loci with deleterious allele frequency q per locus. From this population we randomly sample N parents. This individual sampling can be expressed by two random variables - number of heterozygous (v_{1i}) and homozygous (w_{1i}) loci of the i-th parent for $i = 1, ...N$, for the lethal alleles. Next, we do selfing for each of these parents to obtain the selfed offspring of the first generation, $n_{1i}, i = 1, ...N$. Next, we check the number of unviable offspring, d_{1i}, for each family (parent). The genetic mortality of each offspring from the i-th parent can be represented as a random variable with the given parameter values v_{1i}, w_{1i}, which are the realizations of the first sampling distribution. The second generation selfing is the same as in the first generation except that we randomly choose one viable offspring from each family line, and proceed to a second generation of selfing. The sampling and mortality distributions in the second generation can be defined as n_{2i}, d_{2i}, v_{2i}, and w_{2i} as in the first generation. Family lines can be lost when no viable offspring is obtained in the first generation (Figure 1).

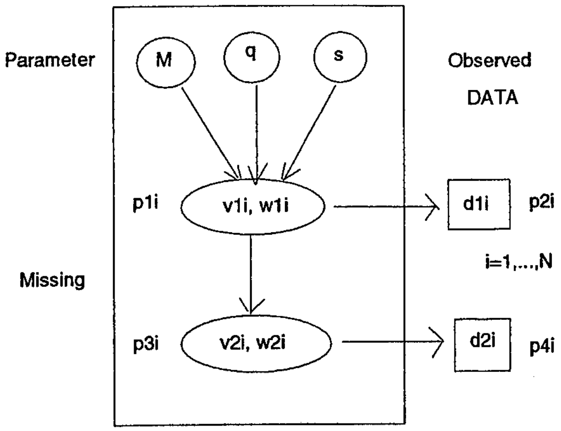


Figure 1: Hierarchical structure of two generation mating experiment and missing data

If we assume that all the loci with lethal alleles act independently, we can define the sampling distribution of parents (P_{1i}) by a multinomial distribution with parameters M, total lethal loci, $p_1 = \frac{2q(1-q)}{1-sq^2}$, and $p_2 = \frac{q^2(1-s)}{1-sq^2}$, relative frequencies of homozygous and heterozygous genotypes, respectively. The sampling distribution of the second generation (P_{3i}) conditional upon the parent lethal counts is a function of the lethal counts v_{1i}, w_{1i} and selection coefficient s, but not of parameters M and q because if a parent having (v_{1i}, w_{1i}) lethal loci is chosen, no further information about mortality of the offspring is gained by knowing M and q; the relative frequencies of homozygous and heterozygous individuals after viable selection is a function of both numbers of deleterious loci of the parent and selection coefficient (Falconer, 1981). P_{3i} can also be derived as another multinomial distribution with parameters v_{1i}, $p_3 = \frac{1/2}{(1-s/4)}, p_4 = \frac{(1-s)/4}{(1-s/4)}$. Note that homozygous loci will be transmitted as homozygous to offspring by selfing. If we assume that the viability of each progeny from one parent is conditionally independent upon a given parent genotype, (v_{1i}, w_{1i}) at the first generation (or (v_{2i}, w_{2i}) at the second generation), the mortality distributions of the two generations, P_{2i}, P_{4i}, are binomial trials with parameters n_{ki}, $Q_{ki} = 1 - (1 - \frac{s}{4})^{v_{ki}}(1 - s)^{w_{ki}}$, $k = 1, 2$, $i = 1, ..., N$, respectively. So, the complete likelihood including missing data can be derived as the product of these four distributions over all N families: $L_c(M, q, s : \{v_{1i}, w_{1i}, v_{2i}, w_{2i}\})$

$$= \prod_{i=1}^{N} P_{1i}P_{2i}(P_{3i}P_{4i})^{U_i}, \qquad (2)$$

where U_i is the indicator function whether family i is extinct or not at the first generation. However, because we do not observe lethal loci counts, v_{ki}, w_{ki}, $k = 1, 2$, $i = 1, ..., N$, equation (2) is not the actual likelihood of the parameters. Importantly, the actual likelihood $L(M, q, s)$ is a summation over all possible configurations of missing data in equation (2).

3 Markov Chain Monte Carlo

3.1 Strategy

The Markov chain Monte Carlo (MCMC) method (Gelfand and Smith 1990, Smith and Roberts 1993, Besag and Green 1993) method is used to overcome the fact that the actual likelihood is analytically intractable. Our strategy is to apply Bayesian analysis by formulating a prior distribution $\pi_0(M, q, s)$ over the parameter space and then using MCMC to simulate the joint posterior distribution of parameters and missing data. This distribution has density π

$$\pi \propto L_c(M, q, s : \{v_{1i}, w_{1i}, v_{2i}, w_{2i}\})\pi_0(M, q, s)$$

and can be readily evaluated (up to constant). We use marginal posterior distributions as the basis for inference.

So, our complete likelihood L_c is defined over a $3 + 4\sum_{i=1}^{N} n_i$ dimensional space of parameters and missing data, $(M, q, s, \boldsymbol{v_1}, \boldsymbol{w_1}, \boldsymbol{v_2}, \boldsymbol{w_2})$. We, first, consider independent uniform priors on the three parameters; independent uniform (0,1) priors are used for the lethal allele frequency and selection coefficient; we consider a discrete uniform prior on M between 100 and 10,000 because the total number of lethal loci of natural species is known to be bounded by a certain number.

The MCMC algorithm has 3 component chains - each modifying different aspects of the state $x = (M, q, s, v_{1i}, w_{1i}, v_{2i}, w_{2i})$. Each component chain is defined by a proposal distribution

$$q(x, x^*) \tag{3}$$

that says how candidate state are generated given current state. A move to x^* occurs with probability that is the minimum of 1 and the Metropois-Hastings (MH) ratio

$$r = \frac{\pi(x^*)}{\pi(x)} \frac{q(x^*, x)}{q(x, x^*)} \tag{4}$$

These 3 components allow us to update each state value in the situation that a direct update is intractable.

3.2 Missing Data Update

A Metropolis-Hastings (MH) algorithm for updating each of the missing data values, v_{1i}, w_{1i}, v_{2i}, and w_{2i} is used. We use the sampling distributions, P_{1i}, P_{3i}, in the hierarchical model as the proposal distributions of the MH algorithm. For instance, the proposal distribution of v_{1i} can be derived as a binomial distribution with parameters $M - w_{1i}$ and $\frac{p_1}{1-p_2}$, where p_1 and p_2 are defined as in the previous section. Then, we sample a candidate v_{1i}^* from the proposal distribution, and calculate the MH ratio in equation (4) to decide whether we move to the new state value v_{1i}^*. The other missing data can be updated in a similar manner to this.

4 Parameter Updates

Choice of proposal distribution in the MH algorithm can dramatically affect efficiency. For example, proposals which tend to be close to the current state yield high acceptance rates but may lead to slowly mixing chains. Similarly, overly disperse proposals will move far but have low acceptance rates. A good proposal distribution balances these opposing constraints.

For updating s, we use a random walk proposal distribution, which chooses a candidate uniformly from a neighborhood of the current state value of s. Specifically, we choose a candidate s^* uniformly from a small interval with length 2δ around the current state value s, and, then, decide whether to move to s^* or not by calculating the MH ratio of s^* and s in equation (4). The width of the neighborhood can be determined by considering the acceptance ratio of the proposal chain; the wider the width, the lower the acceptance ratio. However, if it is too narrow, the chain also mixes slowly because a candidate cannot move far from the current value. So, we need to compromise. This proposal chain can be constructed to be symmetric by adopting an interval width 2δ from an edge if the current value of s is close to the boundary. Thus, the transition probabilities of the proposal distribution in equation (4) can be canceled out, so that the MH ratio can be reduced to the likelihood ratio of the candidate s^* and the current state value s.

In a preliminary study, the M and q appeared to be highly correlated in their posterior; they are distributed in attenuated region along a reciprocal line (Figure 2). Note that the product of these two parameters should be kept as a constant under the fixed number of lethal equivalents because, basically, the product represents the average number of lethal alleles carried by an individual from the population.

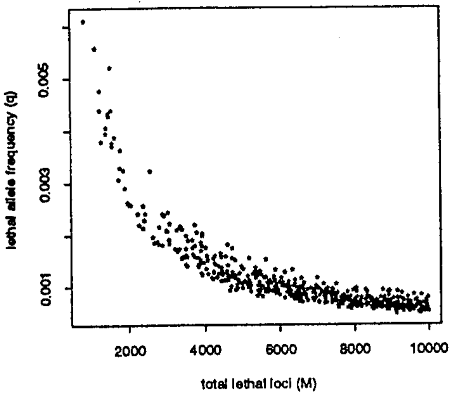


Figure 2: Joint posterior of (M, q)

Such high correlation is well-known to cause problems for any single-site or componentwise updating scheme. By using two single-site proposal distributions, our MCMC implementation mixed extremely slowly (data not shown). To overcome this problem, we use a a two-dimensional proposal distribution based on the knowledge that lines of constant Mq will have approximately constant posterior density. This two dimensional proposal chain is not symmetric, and the ratio of the proposal transition probabilities is now $\frac{q((M^*,q^*),(M,q))}{q((M,q),(M^*,q^*))} = \frac{1/M^*}{1/M} = \frac{M}{M^*}$.

Two dimensional MH Proposal distribution

1. sample M^* from a discrete uniform distribution in a bounded interval.
2. for some constant δ, sample q^* uniformly from the interval
   $\frac{Mq-\delta}{M^*} < q^* < \frac{Mq+\delta}{M^*}$
   where M, q are current values.
3. calculate the two-dimensional MH ratio
   $r = \frac{\pi(M^*,q^*|rest)}{\pi(M,q|rest)} \frac{M}{M^*}$.
4. move to (M^*, q^*) with probability $\min(r,1)$.

Note that, in step 2, if we are close to a boundary of the support of q, the interval with length $\frac{2\delta}{M^*}$ is chosen from the edge, and not centered at the current state value q.

Intuitively, our update method does not reparametrize the parameters M, q in the model, but construct a proposal distribution that can move along a reparametrized space of them.

5 Estimation and Marginal density

We apply MCMC to a data set simulated under parameter values $M = 3000, q = .002$, and $s = .45$ (so, $E = 2Mqs = 5.4$). Using 1,500 burn-in time and every tenth subsampling, we got 500 samples of $(M, q, s, v_1, w_1, v_2, w_2)$ from the MCMC run. The posterior means of s and E were estimated closed to the simulated parameter values of them (Table 1).

Table 1. Estimates of posterior samples

	$E(.\|data)$	std. dev.$(.\|data)$
sel. coef (s)	.471	.059
Lethal equiv.(E)	5.55	.35
lethal loci (M)	6592.1	2328.4
allele freq. (q)	1.12e-3	.76e-3

Our two generation mating data do not give a separate information about parameters M and q. As shown in figure 2, their posterior samples were attenuated, and either mean of the two has a large sample variation of the posterior samples.

In this problem, we have two different sources of error - how close the true posterior means are to the parameter (3000,.002,.45) and how close the Monte Carlo estimates are to the true posterior means. The error by the MCMC estimates is different from the distance between a posterior mean and a parameter value. So, we need a little of caution to interpret the estimates.

Figure 3 shows the (posterior) marginal density of s by using Rao-Blackwellization (Gelfand and Smith, 1990). It is highly concentrated around the posterior mean of s.

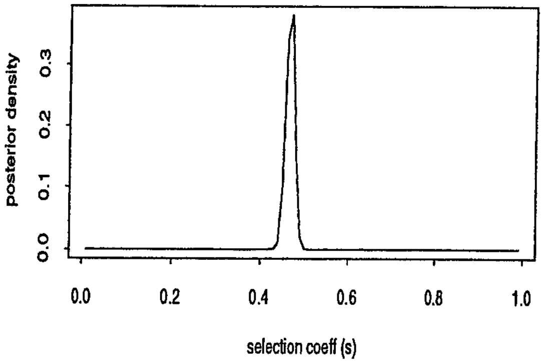


Figure 3: Marginal posterior density of selection coefficient using Rao-Blackwellization

6 Discussion

Our two generation data allow to get information about selection coefficient and to estimate the number of lethal equivalents as well. In our MCMC implementation, the two-dimensional proposal distribution drastically improves efficiency of the algorithm, and overcomes a typical difficulty of single-site or componentwise updating scheme for highly correlated parameters.

Our data and likelihood model do not give information about parameters M and q except highly attenuated joint distribution of them. This seems to be directly related to identifiability problem of likehood for the parameters. We also need to relax many restrictions, e.g., single allele frequency, single selection coefficient, and recessive lethal allele assumptions, which requires to investigate both different statistical modeling and experimental strategy. The difficulty may be turned by using a combination of different mating systems.

References

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation. *J. R. Statist. Soc. B*, **55**, *No.* 1, 25-37.

Falconer, D. S. (1981). *Introduction to quantitative genetics - 2nd ed.* New York: Longman.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *J. Am. Statist. Ass.*, **85**, 398-409.

Lee, J., , Nordheim, E. V., and Kang, H. (1994). Inference for lethal gene estimation. *submitted to Biometrics.*

Morton, N. E., Crow, J. F., and Muller, H. J. (1956). An estimation of the mutational damage in man from data on consanguineous marriages. *Proceeding National Academy of Sciences, USA* **42**, 855-863.

Smith, A. M. F. and Roberts, G. O. (1993). Bayesian computation via Gibbs sampler and related Markov chain Monte Carlo methods. *J. R. Statist. Soc. B* **55**, 3-23.

RELATIVE AGGREGATION AND RANDOM QUADRAT SAMPLING

By C. Cheng and M. A. Johnson
Upjohn Laboratories, Kalamazoo, MI 49007, USA

ABSTRACT.

A distributional parameter, the relative aggregation coefficient (RAC), is derived from a probablistic model of random quadrat sampling. New insight of random quadrat sampling is obtained from the derivation. A firm theoretical ground for the interpretation and the uses of RAC in several statistical applications is established.

1. INTRODUCTION

Random quadrat sampling is a tool often used in statistical ecology (Pielou 1977, Lloyd 1967, Ripley 1981, and Cressie 1991) to study spatial distribution of individuals. A quadrat is a small neighborhood used as a sampling unit. The number of observed individuals in each quadrat, the quadrat counts, are the primary data for inference. A probabilistic model for random quadrat sampling is introduced here. The observed individuals are regarded as i.i.d. realizations from a population distribution P_f having a density f, and the positions of the quadrats are regarded as i.i.d. observations from a design distribution P_g having a density g.

Let μ be the mean of the quadrat counts, and let A be the variance-to-mean ratio of the quadrat counts. It is shown that under a proper condition of the population density f, A is approximately a linear function of μ: $A \approx 1 + [\alpha(f|g) - 1]\mu$, where the slope is determined by the *relative aggregation coeffieicnt* (RAC) of the population density f with respect to the design density g, $\alpha(f|g) := \int f^2 g / (\int fg)^2$. The derivation reveals new insights and uses of random quadrat sampling.

The population density f can be explored using the RAC $\alpha(f|g)$ with different choices of g. It is shown that f is the uniform density on a finite region if and only if the RAC $\alpha(f|g) = 1$ for any density g (f itself in particular) totally concentrating on the same region. The equivalence of two densities is characterized by equalities among six particular RACs. These characterizations provide a firm theoretical ground for RAC-based inferences by random quadrat sampling.

2. QUADRAT SAMPLING IN PROXIMITY SPACES

A *proximity space* is a pair (D, d), where D is a point set, and d is a non-negative real function on $D \times D$ such that $d(x, y) = d(y, x)$, $x, y \in D$. The space $\mathbb{R}^k$ with any ℓ^p metric $(0 < p \leq \infty)$ forms a proximity space $(\mathbb{R}^k, \ell^p)$. For an example of random quadrat sampling in the Euclidean space $\mathbb{R}^2$, see Cressie (1991) pp.588–591. To accommodate applications with non-Euclidean data (see Johnson 1989, Johnson and Maggiora 1990 for examples), the concepts are developed for general proximity spaces.

For convenience of discussion, the proximity function d is regarded as a distance (the larger $d(x, y)$, the further apart are x and y). For any $x \in D$, a "quadrat" at x is simply a neighborhood $B_r(x) = \{y \in D : d(x, y) < r\}$, $r > 0$. Note $B_r(z)$ monotonically decreases as $r \downarrow 0$. Let $(D, \mathcal{D}, \nu)$ be a measure space, where $\mathcal{D}$ is a sigma algebra rich enough so that all the open neighborhoods are $\mathcal{D}$-measurable, and ν is a complete and sigma-finite measure.

REMARK 2.1. Given a proximity measure d, a topology $\mathcal{T}$ can be generated using the open neighborhoods $B_r(x)$, $x \in D$, $r \geq 0$ as basis. Then $\mathcal{D}$ is the smallest sigma algebra generated by the open sets of $\mathcal{T}$.

In the probabilistic model of random quadrat sampling, the data points (e.g. positions of plants in a field under study) are regarded as a random sample $X_1, X_2, ..., X_N$ from a population distribution P_f on $(D, \mathcal{D})$, $P_f << \nu$ with the population density $f = dP_f/d\nu$. The positions of the quadrats are regarded as a random sample from a *design distribution* $P_g << \nu$ with the design density $g = dP_g/d\nu$. Let $Z \sim P_g$. Define the *quadrat count* for $r \geq 0$ at Z to be the random variable

$$\mathcal{N}_r = [\text{number of } X_i\text{'s in the quadrat } B_r(Z)].$$

Define the mean quadrat count $\mu_r := \mathrm{E}(\mathcal{N}_r)$, and the quadrat count variance-to-mean ratio $A_r :=$

$\mathrm{Var}(\mathcal{N}_r)/\mu_r$. An approximate linear relationship between A_r and μ_r can be derived under the condition that as $r \longrightarrow 0$, *a.e.* ν

$$\int_{B_r(z)} f(t)d\nu(t) = h_f(z)\xi_f(r) + o(\xi_f(r)). \quad (2.1)$$

The functions h_f and ξ_f are assumed to depend on the population density f and the proximity measure d (cf. Examples 2.1–2.3 below). For details see Cheng and Johnson (1994a).

Theorem 2.1. *Under condition* (2.1), *as* $r \longrightarrow 0$,

$$A_r = 1 + \left[\frac{N-1}{N}A(f|g) - 1\right]\mu_r + o(\mu_r), \quad (2.2)$$

where

$$A(f|g) = \frac{\int_D h_f^2(t)g(t)d\nu(t)}{\left[\int_D h_f(t)g(t)d\nu(t)\right]^2}. \quad (2.3)$$

$A(f|g)$ will be referred as the quasi relative aggregation coefficient (quasi RAC) of f with respect to g. If h_f coincides with f (see e.g. Example 2.1), the quasi RAC becomes the *relative aggregation coefficient* (RAC)

$$\alpha(f|g) = \frac{\int_D f^2(t)g(t)d\nu(t)}{\left[\int_D f(t)g(t)d\nu(t)\right]^2}. \quad (2.4)$$

The special case $g = f$ gives a new distributional characteristic $\alpha(f) := \alpha(f|f) = \int_D f^3 \Big/ \left(\int_D f^2\right)^2$; call it the self-aggregation coefficient of f.

Condition (2.1) demonstrates the interplay between the proximity measure d and the population density f that leads to the approximate linear relationship (2.2). It is instructive to consider the following examples.

EXAMPLE 2.1. In the Euclidean space $(\mathbb{R}^k, \ell^2)$, if f is at least twice continuously differentiable with bounded mixed derivatives, then Taylor expansion establishes $\int_{B_r(z)} f(t)dt = f(z)v_r + o(v_r)$, $r \longrightarrow 0$, where v_r is the volume of $B_r(z)$. Note when d is the Euclidean distance, $B_r(z)$ is a hyperball of radius r centered at z. In fact condition (2.1) is satisfyed for any ℓ^p distance $(p = 1, 2, ..., \infty)$, if f is at least twice differentiable. In this case the function $h_f = f$ in (2.1).

EXAMPLE 2.2. Consider the truncated bivariate normal density

$f(x_1, x_2) = 4(2\pi)^{-1}\exp\{-(x_1^2+x_2^2)/2\}, x_1, x_2 \geq 0$ on $(\mathbb{R}^2, \ell^2)$. f is infinitely continuously differentiable with bounded mixed derivatives except on an edge ($x_1 = 0$ or $x_2 = 0$) with zero measure. Set the design density $g = f$. Then by Example 2.1, $A(f|g) = A(f|f) = \int f^3/(\int f^2)^2 = 4/3$.

EXAMPLE 2.3. Let f and g be the same as in Example 2.2, but now let d be the proximity $d(\mathbf{x}, \mathbf{y}) = 1 - \left|\mathbf{x}'\mathbf{y}/(\|\mathbf{x}\| \cdot \|\mathbf{y}\|)\right|$, where the vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$, and $\|\mathbf{x}\|$ denotes the length of $\mathbf{x}$. Note $d(\mathbf{x}, \mathbf{y})$ is simply one minus the absolute correlation coefficient between $\mathbf{x}$ and $\mathbf{y}$. It is convenient to use polar coordinates in this case. For a point $\mathbf{z} \in \mathbb{R}^2$ with polar coordinates (ρ_z, θ_z), and $0 \leq r \leq 1$, the quadrat $B_r(\mathbf{z}) = \{\mathbf{x} : d(\mathbf{x}, \mathbf{z}) \leq r\}$ can be represented in polar coordinates as $B_\delta(\mathbf{z}) = \{(\rho, \theta) : \rho \geq 0,\ |\theta - \theta_z| \leq \delta\}$, with $\delta = \arccos(1 - r)$. Note $B_\delta(\mathbf{z})$ is a cone. In polar coordinates the truncated normal density f reads $f(\rho, \theta) = 4(2\pi)^{-1}\rho\exp\{-\rho^2/2\}$, $\rho \geq 0$, $0 \leq \theta \leq 2/\pi$. Thus $\int_{B_\delta(\mathbf{z})} f(\rho, \theta)d\rho d\theta =$

$\int_0^\infty \int_{\theta_z-\delta}^{\theta_z+\delta} 4(2\pi)^{-1}\rho\exp\{-\rho^2/2\}d\rho d\theta = 2\pi^{-1}\delta.$

So condition (2.1) holds with $h_f(\mathbf{z}) \equiv 1$ and $\xi_f(\delta) = 2\delta/\pi$, giving $A(f|g) = \int h_f^2 f/(\int h_f f)^2 = \int f/(\int f)^2 = 1$.

These latter two examples demonstrate that the formation of the quadrats (determined by the proximity d) is essential to what may be observed from the quadrat counts.

The integral $\int_{B_r(z)} f$ represents the local population density in the quadrat $B_r(z)$. When (2.1) holds, the discrepancy between the population density f and the function h_f on the right-hand side represents the possible distortion introduced by quadrat counts in reflecting the true population density. When h_f is constant, the quadrat counts reflect essentially uniformity in the population.

Theorem 2.2. *Under condition* (2.1), $h_f = c$ *(a constant) a.e.* P_g *if and only if the quasi RAC* $A(f|g) = 1$.

Let Z be a random variable following distribution P_g. Then $A(f|g) = \mathrm{E}[h_f^2(Z)]/(\mathrm{E}[h_f(Z)])^2 \geq 1$. The uniformity reflected by quadrat sampling is characterized by $A(f|g)$ attaining the lower bound 1. Similarly, the uniformity in the population relative to a reference density g is characterized by the RAC $\alpha(f|g) = 1$. Define the support of a density f on D to be set $\mathrm{supp}\, f := \{x \in D : f(x) > 0\}$.

A general characterization of uniformity is given by

Theorem 2.3. RAC Characterization of uniformity. *Let f and g be two densities on D with respective supports S_f and S_g. Assume $\nu(S_f \cap S_g) > 0$. Then $\nu(S_g) < \infty$ and $f = c$ (a positive constant) a.e. ν on S_g, if and only if $\alpha(f|g) = 1$.*

Given a region $R \in \mathcal{D}$ with $0 < \nu(R) < \infty$, the uniform density can be defined as $u(x) = [\nu(R)]^{-1} I_R(x)$, $x \in D$, with I the indicator function. The special case $g = f$ gives

Corollary 2.4. *Let f be a density with support S_f. Then $\nu(S_f) < \infty$ and f is a uniform density if and only if $\alpha(f) = 1$.*

Since $\alpha(f|g) \geq 1$, uniformity is characterized by certain RACs attaining the lower bound 1.

Thereom 2.3 reveals the interpretation of the RAC $\alpha(f|g)$ – it measures the contrast between the dense and the sparse regions introduced by the population density f *relative to* the design/reference density g. The uniform distribution over any finite region introduces no density contrast relative to any density totally concentrating on that region. For further discussions and examinations of Examples 2.2 and 2.3, see Cheng and Johnson (1994a).

REMARK 2.2. Note that although the RAC $\alpha(f|g)$ is derived for the random quadrat sampling model which involves a proximity measure d, the definition of RAC requires nothing but two probability densities. Theorem 2.1 demonstrates that if a proximity measure interacts properly with the population density f in the random quadrat sampling, one can arrive at the RAC as a result, and any proximity measure satisfying (2.1) with $h_f = f$ will do. In general, if another proximity measure d' generates a sub-sigma algebra $\mathcal{D}'$ of $\mathcal{D}$ (possibly $\mathcal{D}' = \mathcal{D}$) in the way described in Remark 2.1, then all the quadrats formed by d' are $\mathcal{D}$-measurable. Quadrat sampling can be performed using d' as well, but the results may or may not agree with those from the sampling performed with d, as show in Examples 2.2 and 2.3. The use of quadrat sampling to estimate an RAC will be briefly discussed in Section 4.

3. FURTHER PROPERTIES OF RAC

Further properties of RAC are highlighted here. These properties establish a theoretical ground for the use and interpretation of RAC in statistical inferences using random quadrat sampling. Detailed discussions and elaborations appear in Cheng and Johnson (1994a, b).

Let $(D, \mathcal{D}, \nu)$ be a measure space with ν a complete and sigma-finite measure. A ν-density function on D is a non-negative real function f satisfying $\int_D f d\nu = 1$.

The RAC $\alpha(f|g)$ possesses interesting invariance properties. For example, it is invariant under lineat transforms.

Theorem 3.1. *Let f and g be two densities on $\mathbb{R}^k$ with $\int fg > 0$. Let T be a $k \times k$ nonsingular matrix, and $J = \det(T^{-1})$. Fix $x_0 \in \mathbb{R}^k$. Let $\hat{f}(x) = f(T^{-1}(x - x_0))|J|$, and $\hat{g}(x) = g(T^{-1}(x - x_0))|J|$. Then $\alpha(\hat{f}|\hat{g}) = \alpha(f|g)$. In particular, $\alpha(\hat{f}) = \alpha(f)$.*

Theorem 3.2. Independence. *Let f and g be two densities on $\mathbb{R}^k$ with $\int fg > 0$. If f and g are such that $f = \prod_{i=1}^m f_i$, $g = \prod_{i=1}^m g_i$, where f_i and g_i are densities on $\mathbb{R}^{k_i}$, with $\int f_i g_i > 0$, $\sum_{i=1}^m k_i = k$, then $\alpha(f|g) = \prod_{i=1}^m \alpha(f_i|g_i)$. In particular, $\alpha(f) = \prod_{i=1}^m \alpha(f_i)$.*

Let f_1, f_2 be two densities on D, and let $f = (f_1 + f_2)/2$, i.e., the density of the even mixture.

Theorem 3.3. *$\alpha(f_1|f) \geq \alpha(f|f_1)$ if and only if $\int f_1^3 \geq \int f_2^2 f_1$. $\alpha(f_2|f) \geq \alpha(f|f_2)$ if and only if $\int f_2^3 \geq \int f_1^2 f_2$.*

By rewriting the inequality $\int f_1^3 \geq \int f_2^2 f_1$ as $\int f_1^2 f_1 \geq \int f_2^2 f_1$, it is seen that the inequality $\alpha(f_1|f) \geq \alpha(f|f_1)$ reflects the discrepancy in the concentration of the two densities. The following theorem demonstrates the equivalence of two densities is characterized by equalities among certain RACs.

Theorem 3.4. *$f_1 = f_2$ a.e. ν if and only if $\alpha(f_1) = \alpha(f_2)$, and $\alpha(f_1|f) = \alpha(f|f_1) = \alpha(f_2|f) = \alpha(f|f_2)$.*

Theorem 3.2 shows the behavior of RAC under independence. As a consequence of Theorem 3.4, independence can be characterized by equalities among six particular RACs. For $i = 1, 2$, let $(D_i, \mathcal{D}_i, \nu_i)$ be a measure space with ν_i a complete and sigma-finite measure, let P_i be a probability measure on $(D_i, \mathcal{D}_i)$, $P_i << \nu_i$, with density $f_i = dP_i/d\nu_i$, and let Y_i be a D_i-valued random variable with distribution P_i. Let P be the joint distribution of (Y_1, Y_2) in the product space $(D_1 \times D_2, \mathcal{D}_1 \bigotimes \mathcal{D}_2, \nu)$, where $\mathcal{D}_1 \bigotimes \mathcal{D}_2$ is the product sigma algebra and $\nu = \nu_1 \times \nu_2$. Assume $P << \nu$ with density $f = dP/d\nu$. By definition, Y_1 and Y_2 are independent if $P = P_1 \times P_2$, or equivalently, $f = f_\Pi$ *a.e.* ν, with $f_\Pi = f_1 f_2$. Let $g := (f + f_\Pi)/2$, the even-mixture density of the joint and the product.

Corollary 3.5. *$\alpha(f|g) \geq \alpha(g|f)$ if and only if $\int f^3 \geq \int f_\Pi^2 f$. $\alpha(f_\Pi|g) \geq \alpha(g|f_\Pi)$ if and only if $\int f_\Pi^3 \geq \int f^2 f_\Pi$. $f = f_\Pi$ a.e. ν if and only if $\alpha(f) = \alpha(f_\Pi)$ and $\alpha(f|g) = \alpha(g|f) = \alpha(f_\Pi|g) = \alpha(g|f_\Pi)$.*

A general measure of dependence can be constructed by combining the above RACs.

$$\Delta(P_1, P_2) := \left[\frac{\alpha(f)}{\alpha(f_\Pi)} - 1\right]^2 + \left[\frac{\alpha(f|g)}{\alpha(g|f)} - 1\right]^2 + \left[\frac{\alpha(f_\Pi|g)}{\alpha(g|f_\Pi)} - 1\right]^2 + \left[\frac{\alpha(f|g)}{\alpha(f_\Pi|g)} - 1\right]^2.$$

Note $\Delta(P_1, P_2) = 0$ if and only if the two distributions are independent; the greater $\Delta(P_1, P_2)$, the stronger the dependence between the two distributions. This RAC measure of dependence reflects the deviation from independence by detecting the discrepancy in the concentration of the joint and the product densities in the sample space. It has the advantage of being extremely general, and the drawback of not reflecting the detailed nature of the dependence. See Johnson *et al.* (1994) for a related measure of dependence calibrated against the correlation coefficient of bivariate normal distributions.

4. ESTIMATION OF RAC

Estimation of the RAC $\alpha(f|g)$ by quadrat sampling is briefly discussed here in general terms. Theorem 2.1 suggests the following quadrat count moment estimator of $\alpha(f|g)$ under the condition (2.1) with $h_f = f$ in a proximity space (D, d).

$$\widehat{\alpha}(f|g) = \frac{N}{N-1}\left(1 + \frac{V_r}{m_r^2} - \frac{1}{m_r}\right), \qquad (4.1)$$

where N is the total number of observed individuals, $r > 0$ is a number close to 0 in condition (2.1), and V_r and m_r are respectively the sample variance and sample mean of the quadrat counts from a sample of size-r quadrats taken from the the design distribution.

Let n be the number of random quadrats. The consistency of moment estimators implies that the $\widehat{\alpha}(f|g)$ convergence in probability as $\min(N, n) \uparrow \infty$ to the function $a(A_r, \mu_r) = 1 + (A_r/\mu_r) - (1/\mu_r)$ with μ_r and A_r the quadrat count mean and variance-to-mean ratio respectively. Under condition (2.1) with $h_f = f$, $a(A_r, \mu_r) = \alpha(f|g) + o(1)$, $r \downarrow 0$, $N \uparrow \infty$, so $\widehat{\alpha}(f|g)$ consistently estimates an approximation of the RAC.

REFERENCES

Cheng, C. and Johnson, M. A. (1994a) Relative aggregation coefficient characterizations: a basis for inference on proximity data using random quadrat sampling. manuscript.

Cheng, C. and Johnson, M. A. (1994b) Relative aggregation coefficients for describing and comparing densities in proximity spaces. manuscript.

Cressie, N. (1991). *Statistics for Spatial Data.* New York, John Wiley & Sons.

Johnson, M. A. (1989). A review and examination of the mathematical spaces underlying molecular similarity analysis. *Journal of Mathematical Chemistry* **3** 117–145.

Johnson, M. A., Cheng, C., Maggiora, G. M. and Lajiness, M. S. (1994) A generalized measure of dependence with an application to molecular similarity analysis. In *Proceedings of Computer Science and Statistics: Interface'94* – this volume.

Johnson, M. A. and Maggiora, G. M. (1990). *Concepts and Applications of Molecular Similarity.* New York, Wiley Inter-Science.

Lloyd, M. (1967). Mean crowding. *Journal of Animal Ecology* **36** 1–30.

Pielou, E. C. (1977). *Mathematical Ecology.* New York, John Wiley & Sons.

Ripley, B. D. (1981). *Spatial Statistics.* New York, John Wiley & Sons.

Efficient Computation of Statistical Procedures based on Subsetting the Observations

John E. Hinkle and Arnold J. Stromberg

Abstract

Many statistical techniques require that computations be done on all subsets of size r in a data set of size n. Typically, this is done lexographically, i.e., with nested do-loops. If an exchange one point update formula is available, then it is used on the inner loop. In this paper we discuss a method of counting through all subsets of size r in a data set of size n by changing only one element as one goes from one subset to the next. The advantage of such methods is that an update formula can be used at every step, thus potentially saving computation time. The method used to compute the next subset in the list requires some computation time, and thus the new method will only be faster if the update formula is sufficienty faster than doing the computation from scratch.

1 Introduction

Statistical procedures such as jackknife estimation, influence diagnostics, cluster analysis, and permutation tests call for computations on all size r subsets of an n element observation dataset. As a result these procedures can be very computer-intensive. Though computational speed is improving, these procedures can easily exceed available resources and therefore, are not considered for use in some applications, thus algorithm efficiency plays an extremely important part in accessing applicability. In this paper we will combine different subset generating methods with iterative updating techniques and build algorithms that minimize the number of floating point operations(FLOPs) necessary for computations. This FLOP minimization, as a result, will expand the situations in which these procedures can be used.

The remainder of this paper will be organized as follows. In section 2 we will describe subsetting procedures. In section 3 we will look at the computation of subsetting procedures based on different subset generating techniques. In section 4 we will discuss the prove of the existence of change-one subset generators and give an algorithm for one such method. In section 5 we will look at the relative efficiency of using different subset generators in general computational procedures that allow for iterative updating.

2 Subsetting Procedures

Computing the statistical procedures mentioned in the introduction requires sequentially generating all $\binom{n}{r}$ subsets of the n observations. If the observations are indexed by the set $\{1,\ldots,n\}$, then let all size r subsets be denoted by $S_{n,r}=\{s_1,s_2,\ldots,s_{\binom{n}{r}}\}$ where $s_k=\{i_1,\ldots,i_r\}$ and $1\le r\le n$. Using this set of indice subsets we could, for example, compute Delete-d Cook's Distance,

$$D_{s_k}\propto\left(b-b_{s_k}\right)'\left(X'X\right)\left(b-b_{s_k}\right),\qquad(1)$$

by sequentially counting through the subsets. Cook's Distance measures the influence that subset $(Y_{s_k},X_{s_k})=\{(Y_i,X_i):i\in s_k\}$ has on the regression model

$$Y_{n\times1}=X_{n\times p}\beta_{p\times1}+\varepsilon_{n\times1},\qquad(2)$$

where b is an estimate of β, and b_{s_k} is an estimate of β based on the size $r=n-d$ subset indexed by s_k. Clearly the computer intensive part of this procedure is in computing b_{s_k} for each subset. Another example is the Delete-d Jackknife variance estimation,

$$v_{J,n-d}\left(\hat{g}\right)\propto\sum_{s_k\in S_{n,n-d}}\left|X'_{s_k}X_{s_k}\right|\left(\hat{g}-\hat{g}_{s_k}\right)\left(\hat{g}-\hat{g}_{s_k}\right)',\qquad(3)$$

given by Wu[1] for the regression model in (2). Here $\hat{g}=\hat{g}\left(b\right)$ is a smooth function of the estimated regression coefficients. The computer intensive aspect is in calculating b_{s_k} and $\left|X'_{s_k}X_{s_k}\right|$ for all subsets.

The examples given above can be put into a general context which we will call *Subsetting Procedures.* A subsetting procedure is an aggregate calculation or operation involving a basic function of each specified size subset of the observations of interest. For instance, Cook's Distance calculates regression coefficients for each $r=n-d$ size subset of the observation with the aggregate operation being a list or partial list of the influence measure of each subset. Likewise, Wu's jackknife variance estimation calculates a function of the observations for each subset with the aggregate being the sequential sum over all the subsets.

So, based on these examples, every subsetting procedure will have a basic function or operation that is repeatedly calculated. If this basic operation is called θ, then with the observations given by $X = \{x_1, \ldots, x_n\}$ and the subset by $X_{s_k} = \{X_i : i \in s_k\}$, a subsetting procedure will calculate $\theta_{s_k} = \theta(X_{s_k})$ for each $s_k \in S_{n,d}$. Depending on the ultimate calculation involved, computing θ_{s_k} will usually involve part of the overall computations on the current subset. In Delete-d Cook's Distance the basic operation would be calculating the regression coefficients while the overall computation for each subset is the influence measure based on those coefficients. Hereupon we assume evaluating θ_{s_k} represents the bulk of calculations made involving the k'th subset in the subsetting procedure.

3 Computing

Calculating a subsetting procedure is easily accomplished using the following algorithm,

```
for $k = 1$ to $\binom{n}{r}$
  generate subset $s_k$
  $\theta_{s_k} = \theta(X_{s_k})$
end.                                         (4)
$f = f\left(\theta_{s_1}, \ldots, \theta_{s_{\binom{n}{r}}}\right)$
```

The term f above is the aggregating operation of the subsetting procedure, this term will be suppressed in subsequent algorithms. The question now is how to generate the subsets. Usually the subsets are generated lexicographically or alphabetically. An algorithm for computing the subsetting procedure in (4), generating $S_{n,3}$ lexicographically, follows.

```
$k = 1$
for $i_1 = 1$ to $n - 2$,
    for $i_2 = i_1 + 1$ to $n - 1$,
        for $i_3 = i_2 + 1$ to $n$,
            $s_k = \{i_1, i_2, i_3\}$
            $\theta_{s_k} = \theta(X_{s_k})$          (5)
            $k = k + 1$
        end
    end
end.
```

For the above algorithm if we look at the subset generation alone, we have for $n = 5$,

$$S_{5,3} = \left\{ \begin{array}{ll} s_1 = \{1,2,3\}, & s_6 = \{1,4,5\}, \\ s_2 = \{1,2,4\}, & s_7 = \{2,3,4\}, \\ s_3 = \{1,2,5\}, & s_8 = \{2,3,5\}, \\ s_4 = \{1,3,4\}, & s_9 = \{2,4,5\}, \\ s_5 = \{1,3,5\}, & s_{10} = \{3,4,5\} \end{array} \right\}.$$

The algorithm in (5) is straight forward and easy to code for any size problem, but depending on the procedure involved, dataset size and subset size this can be a formidable task! To lessen this cost, we need to improve computational efficiency. If we assume that the formula or logical structure in computing θ_{s_k} is already efficient then the only other choice we have is to use an iterative updating scheme that relies on the results or partial results of computing $\theta_{s_{k-1}}$. So, to make the above algorithm more computationally efficient suppose we can calculate θ_{s_k} by updating $\theta_{s_{k-1}}$. The following code is based on using an update inside the inner loop of the algorithm (5).

```
$k = 1$
for $i_1 = 1$ to $n - 2$,
    for $i_2 = i_1 + 1$ to $n - 1$,
        $s_k = \{i_1, i_2, i_2 + 1\}$
        $\theta_{s_k} = \theta(X_{s_k})$
        $k = k + 1$
        for $i_3 = i_2 + 2$ to $n$,
            $s_k = \{i_1, i_2, i_3\}$
            $\theta_{s_k} = \text{Update}(\theta_{s_{k-1}}, X_{s_k}, X_{i_3}, X_{i_3 - 1})$
            $k = k + 1$
        end
    end
end.
                                                    (6)
```

Notice the difference between this code and the code given in (5). The update formula relies on the single element change made in the subset during the inner loop of the algorithm. If we suppose that change-one generated subsets allow for more efficient computation of subsetting procedures and we have a change-one type of update then we would want to use the update more often during the calculations. Code for such a procedure would have the form,

```
$\theta_{s_1} = \theta(s_1)$
for $k = 2$ to $\binom{n}{r}$
  generate subset $s_k$ by
  changing one element in $s_{k-1}$          (7)
  $\theta_{s_k} = \text{Update}(\theta_{s_{k-1}}, s_k)$
end.
```

4 Change-One Generator

The use of iterative updates in scientific computing is well established. In our case since we are iterating through a sequence of subsets then, as we did above, placing the update where there is a minimal change between consecutive subsets will give a more efficient updating scheme. Assuming our procedure allows for a change-one type updating scheme, does there exist an alternative method of generating subsets that will list all subsets by making single element changes?

Leo W. Lanthroum discovered in 1965 an algorithm that generates $S_{n,r}$ in which a single element is changed in one subset to generate next. Chase[2] presented code for this algorithm and subsequent methods based on a modified Binary Reflected Gray Codes were developed by Nijenhuis and Wilf[3], Bitner et.al.[4], and recently Brezovec and Lee[5]. These methods will be called change-one(C1) subset generators throughout the remainder of this paper. To illustrate, $S_{5,3}$ when generated by a change-one generator becomes,

$$\mathrm{C1}\left[S_{5,3}\right] = \left\{ \begin{array}{ll} s_1 = \{1,2,3\}, & s_6 = \{2,4,5\}, \\ s_2 = \{1,3,4\}, & s_7 = \{3,4,5\}, \\ s_3 = \{2,3,4\}, & s_8 = \{1,3,5\}, \\ s_4 = \{1,2,4\}, & s_9 = \{2,3,5\}, \\ s_5 = \{1,4,5\}, & s_{10} = \{1,2,5\} \end{array} \right\}.$$

Notice that one and only one change is made in going from one subset to the next. By using only the changing element, $\mathrm{C1}\left[S_{n,r}\right]$ can also be described by an initial subset $s_{I,n,r}$ and the set of ordered pairs, $s^*_k = (\mathsf{out}_k, \mathsf{in}_k)$, where in_k is the indice of the element entering the k'th combination and out_k is the exiting indice. The resulting list of order pairs, call it $S^*_{n,r} = \{s_{I,n,r}, s^*_2, s^*_3, \ldots, s^*_{\binom{n}{r}}\}$, will be for $\mathrm{C1}[S_{5,3}]$,

$$S^*_{5,3} = \left\{ \begin{array}{rl} s_{I,5,3} = \{1,2,3\}, & s^*_6 = \{1,2\}, \\ s^*_2 = \{2,4\}, & s^*_7 = \{2,3\}, \\ s^*_3 = \{1,2\}, & s^*_8 = \{4,1\}, \\ s^*_4 = \{3,1\}, & s^*_9 = \{1,2\}, \\ s^*_5 = \{2,5\}, & s^*_{10} = \{3,1\} \end{array} \right\}.$$

The method for constructing and proving the existence of change-one subset generators is given in Nijenhuis and Wilf([3, 1975]). Their method can be described, easily, by letting $S_{n,r} = \left\{s_1, s_2, \ldots, s_{\binom{n}{r}}\right\}$ be a change-one generated subset list with $s_1 = \{1,2,\ldots,r\}$ and $s_{\binom{n}{r}} = \{1,2,\ldots,r-1,n\}$. Then if $\overline{S_{n,r}}$ represents $S_{n,r}$ but in reverse order, we have

$$S_{n,r} = \left\{ \begin{array}{l} S_{n-1,r} \quad , \\ \\ S_{n-2,r-2} \cup \{n-1,n\}, \\ \\ \overline{S_{n-2,r-1}} \cup \{n\}. \end{array} \right\} \qquad (8)$$

Thus for any n and r, where $0 \leq r \leq n$, it follows that $S_{n,r}$ exists. A point of interest is that this method will build any subset list independent of how the minor subsets $S_{n-1,r}, S_{n-2,r-2}$, and $S_{n-2,r-1}$ were generated. For our case, assuming the minor subsets are change-one generated, the result is a change-one list. The proof is by induction.

An algorithm that actually generates a change-one subset list is more complicated than the description given in (8). The algorithm described below is by Brezovec and Lee[5].

4.1 A Change-One Subset Generating algorithm

One method of generating a change-one subset list $S_{n,r}$ is to let $s_1 = \{1,\ldots,r\}$ and then go from $s_k = \{i_1, \ldots i_r\}$ to s_{k+1}, by first determining

$$q = \min\left(k : d_k > 0\right) \qquad (9)$$

where

$$d_k = \begin{cases} n - i_r & \text{if } k = r, \\ i_{k+1} - i_k - 1 & \text{if } i < r \text{ and } r - k \text{ is even,} \\ i_k - k & \text{if } r - k \text{ is odd.} \end{cases}$$

Then setting $s_{k+1} = \{i'_1, \ldots, i'_r\}$ where,

$$i'_k = \begin{cases} i_k & \text{if } k \in \{q+1, \ldots, p\}, \\ i_q + (-1)^{r-q} & \text{if } k = q, \\ i'_k - 1 & \text{if } k = q-1 \text{ and } r-k \text{ is odd,} \\ k - 1 & \text{otheriwse.} \end{cases}$$

If no k satisfies (9) then s_k is the last member of the list. In addition, we can generate the list $S^*_{n,r}$, by using the subset s_k and the number q found in (9). Let $s^*_{I,n,r} = \{1, \ldots r\}$, and $s^*_2 = \{r-1, r+1\}$, now setting $s^*_{k+1} = \{\mathsf{out}, \mathsf{in}\}$, where

$$\mathsf{in} \quad = \quad \begin{cases} i_{q-1} & \text{if } q > 1 \text{ and } r - q + 1 \text{ is even} \\ i_q + (-1)^{r-q} & \text{otherwise,} \end{cases}$$

$$\mathsf{out} \quad = \quad \begin{cases} i_{q-1} & \text{if } q > 1 \text{ and } r - q + 1 \text{ is odd} \\ i_q & \text{otherwise.} \end{cases}$$

5 Relative Efficiency

Our goal is to make subsetting procedures more efficient by reducing the computational cost. The specific cost can be execution time, number of floating point operations or both combined. Here we will use the amount of floating point operations (FLOPs) to measure the cost, with the standard convention of counting each addition as one FLOP and each multiplication as one FLOP.

To improve the efficiency of a subsetting procedure we will concentrate on the basic function θ_{s_k} and the subset generator, which will be either change-one or lexicographic. Suppose it takes at least F FLOPs to compute θ_{s_k}, where F is a function of the size of the subset dataset. The total FLOPs for computing θ_{s_k} for all $s_k \in S_{n,r}$ will be at most $\binom{n}{r} \cdot F$. This would be the amount needed to compute a general algorithm like (5) , call this method straight-forward-lexicographic (SFL) . The amount of FLOPs need to compute a procedure based on the inner-loop update (ILU) algorithm (6) is at most $\binom{n-1}{r-1} \cdot F + \binom{n-1}{r} \cdot \gamma \cdot F$, where $\gamma \cdot F$ is the amount of FLOPS needed to compute the update function and $0 \leq \gamma \leq 1$. Here we make the obvious assumption that the update is less costly to compute than the basic function. Additionally, if we base our computations on the change-one algorithm (7), updating θ_{s_k} each time, then the FLOPs needed will be $F + \left[\binom{n}{r} - 1\right] \cdot \gamma \cdot F + \alpha \cdot \binom{n}{r}$ where α is the maximum FLOPs needed to generate an indice subset. Using the above naive FLOP counts to evaluate the efficiency of computing a general subsetting procedures based on the three algorithms presented in section 5, we have the following relative efficiencies.

$$\begin{aligned}
\mathsf{rel}\,(\text{ILU,SFL}) &= \tfrac{r}{n} + \gamma \tfrac{n-r}{n}, & (10)\\
\mathsf{rel}\,(\text{C1,SFL}) &= \gamma + \tfrac{\alpha}{F}, & (11)\\
\mathsf{rel}\,(\text{C1, ILU}) &= \tfrac{n \cdot (\gamma + \frac{\alpha}{F})}{n \cdot \gamma + (1-\gamma) \cdot r}. & (12)
\end{aligned}$$

Our use of relative efficiency (rel) implies that A is more efficient than B if $\mathsf{rel}(A, B) < 1$. Using this definition of relative efficiency we see that a subsetting procedure utilizing a C1 type algorithm will be more efficient than a SFL algorithm if $\alpha \leq F \cdot (1 - \gamma)$, or generating a subset costs less than the saved cost in updating the basic function. Moreover, a C1 algorithm is preferred when,

$$\alpha \leq F\,(1-\gamma)\,\frac{r}{n}, \qquad (13)$$

or the cost of computing a change-one subset is a fraction of the saved cost in updating the basic function.

6 Conclusion

The above discussion relies on the existence of a stable update of the basic function. Finding a stable update may not be a trivial matter when looking for ways to improve a subsetting procedure. This is beyond the scope of this paper. We did show that a subsetting procedure can be improved if an update exists. That is nothing new, but from (13) we see that a C1 algorithm can only improve a subsetting procedure to a limit. The limiting factor being the cost of computing a change-one subset. This is certainly intuitive and it does open the door for further research. This research would be to find a subset generator that requires a minimal amount of FLOPs, hopefully depending on n. The algorithm we gave in section 4 requires at most n FLOPs to generate a subset. This limits the situation in which it will be useful.

References

[1] Wu, C.F.J. (1986) Jackknife, Bootstrap and other Resampling Methods in Regression Analysis. *Ann. Statist.* **14** 1261-1294.

[2] Chase, P.J.(1970) Algorithm 382; Combinations of M Out of N Objects. *Comm. ACM.* **13** 368-369.

[3] Nijehhuis, A. and Wilf, H.S. (1975) *Combinatorial Algorithms.* Academic Press, New York.

[4] Bitner, J.R., Ehrlich, G., and Reingold, E.M. (1976) Efficient Generation of the Binary Reflected Gray Code and Its Applications. *Comm. Acm.* **19** 517-521.

[5] Brezovec, C. and Lee, C.W. (1992) Enumerating Subsets. Unpublished manuscript.

A frequency domain bootstrap for time series

D. Janas and R. Dahlhaus

Institut für Angewandte Mathematik
Universität Heidelberg
Im Neuenheimer Feld 294
69120 Heidelberg, Germany

Abstract. The properties of a bootstrap based on studentized periodogram ordinates are investigated. We give a correction which emulates the dependence structure of the periodogram. Furthermore, we study the case of tapered data.

1. Introduction. In time series analysis there exists no canonical way to bootstrap the observed data set. The reason is that one essentially has only one multivariate observation available. In order to construct a bootstrap one therefore needs additional informations (e.g. on the dependence structure of the process or on the statistic to be bootstrapped). As a consequence a variety of different bootstrap methods have been suggested which have their merits in different situations (cf. Künsch, 1989; Liu and Singh, 1988; Politis and Romano, 1992; Freedman, 1984; Kreiss and Franke, 1989; Hurvich and Zeger, 1988). In this paper we study a frequency bootstrap based on studentized periodogram ordinates. Although, the periodogram ordinates at different frequencies are asymptotically independent (which is the basis for this bootstrap idea - cp. Franke and Härdle, 1992) the minor dependence sums up in certain statistics to a nonvanishing contribution. Thus, an ordinary bootstrap with an independent bootstrap sample does not lead to a valid bootstrap approximation for certain statistics. We therefore suggest in this paper a modification which leads to a dependent frequency domain bootstrap sample.

2. The method. Let X_t, $t \in \mathbb{Z}$ be a real-valued stationary time series with spectral density f and

$$I_T(\alpha) = \frac{1}{2\pi\, H_{2,T}} \left|\sum_{t=1}^{T} h(\tfrac{t}{T})(X_t - \overline{X}) \exp(i\lambda t)\right|^2$$

be the tapered periodogram with the data-taper $h: [0,1] \to \mathbb{R}$ and $H_{k,T} = \sum_{t=1}^{T} h(\frac{t}{T})^k$. We are interested in the distribution of spectral mean estimates, i.e. in the distribution of

(2.1) $$\sqrt{T}(A(\phi,I_T) - A(\phi,f))$$

where

$$A(\phi,f) = \int_0^\pi \phi(\alpha)\, f(\alpha)\, d\alpha.$$

Examples are estimates for the covariance function ($\phi(\alpha) = 2\cos\alpha n$) and the spectral measure ($\phi(\alpha) = \chi_{[0,\lambda]}(\alpha)$). The asymptotic distribution of (2.1) is well known. It is under suitable regularity conditions a Gaussian distribution with mean zero and variance

(2.2) $$v = c_h[2\pi\int_0^\pi \phi^2(\alpha)f^2(\alpha)d\alpha + (\kappa_4/\sigma^4)\left(\int_0^\pi \phi(\alpha)f(\alpha)\, d\alpha\right)^2], \text{ where } c_h = \|h\|_4^4 \,/\, \|h\|_2^4 .$$

(cf. Dahlhaus, 1983). Here it is assumed that X_t is a linear process with innovation sequence ε_t where $\mathrm{var}(\varepsilon_t) = \sigma^2$ and $\mathrm{cum}_4(\varepsilon_t) = \kappa_4$.

We now approximate the distribution of (2.1) by a bootstrap in the frequency domain. The basic idea results from the fact that $I_T(\alpha)/f(\alpha)$ are for different $\alpha \not\equiv 0 \bmod \pi$ asymptotically independent. This suggests the following bootstrap procedure. Let $n = [T/2]$ and $I_j = I_T(\frac{2\pi j}{T})$.

(2.3) Bootstrap procedure

(a) Obtain the sample of periodogram ordinates $\{I_j\}$ for $j = 1, \dots, n$.

(b) Obtain an estimate $\hat{f}_T$ of the spectral density (e.g. a kernel estimate). Let $\{\hat{f}_j\} \equiv \{\hat{f}_T(\frac{2\pi j}{T})\}$.

(c) Calculate the studentized periodogram ordinates $\{\hat{\varepsilon}_j\} \equiv \{I_j/\hat{f}_j\}$.

(d) Rescale $\hat{\varepsilon}_j$ and consider $\{\tilde{\varepsilon}_j\} \equiv \{\hat{\varepsilon}_j/\hat{\varepsilon}_.\}$ where $\hat{\varepsilon}_. = \frac{1}{n}\sum_{j=1}^{n} \hat{\varepsilon}_j$.

(e) Draw independent bootstrap replicates $\{\varepsilon_j^*\}$ from the empirical distribution of $\{\tilde{\varepsilon}_j\}$.

(f) Define the bootstrap periodogram values by $\{I_j^*\} \equiv \{\hat{f}_j\, \varepsilon_j^*\}$.

The rescaling in (d) avoids an unneccessary bias at the resampling stage. We now can approximate the distribution of (2.1) by

(2.4) $$\sqrt{T}(B(\phi,I_T^*) - B(\phi,\hat{f}))$$

This paper will appear in the Proceedings on the Interface '94.

where

$$B(\phi, I_T^*) = \frac{\pi}{n} \sum_{j=1}^{n} \phi_j \, I_j^* .$$

If $\sup_{\alpha} | \hat{f}_T(\alpha) - f(\alpha) | \to 0$ a.s. then we can check under suitable regularity conditions that (2.4) is also asymptotical normal with mean zero and variance

$$(2.5) \qquad 2\pi \int_0^{\pi} \phi^2(\alpha) \, f^2(\alpha) \, d\alpha .$$

A comparison with (2.2) implies that the bootstrap can only be consistent if $c_h = 1$ and $\kappa_4 = 0$.

To get an idea how to improve the above bootstrap we may look more detailed at the correlation of the periodogram at neighbouring periodogram ordinates. By some standard cumulant calculations (cf. Brillinger, 1981) we obtain for $j \neq k \in \{1, \dots, n\}$ in the simplest case $h(x) \equiv 1$, $f(\alpha) = \frac{\sigma^2}{2\pi}$.

$$(2.6) \qquad \operatorname{cov}(I_j, I_k) = f_j \, f_k \, \{\delta_{jk} + (\kappa_4/\sigma^4) \, T^{-1}\}$$

(for arbitrary linear processes and $h(x) \equiv 1$ one can establish the same result with a remainder $O(\frac{\ln^3 T}{T^2})$ if $j \neq k$ and $O(\frac{\ln T}{T})$ if $j = k$). This implies

$$\operatorname{var}(\sqrt{T}(B(\phi, I_T) - B(\phi, f))$$

$$= T \frac{\pi^2}{n^2} \sum_{j=1}^{n} \phi_j^2 \, f_j^2 + (\kappa_4/\sigma^4) \, (\frac{\pi}{n} \sum_{j=1}^{n} \phi_j \, f_j)^2$$

which tends to v as in (2.2) with $c_h = 1$. Therefore, we need a bootstrap sample that fulfills the analogue to (2.6). As shown below this is fulfilled by the following.

(2.7) Modified bootstrap procedure.
(a) – (e) as in the bootstrap procedure (2.3).
(f) Take an estimate $\hat{\eta}_4$ of $\eta_4 := (\kappa_4/\sigma^4)$ and define the bootstrap periodogram values by

$$\{I_j^*\} = \{\hat{f}_j \, [\varepsilon_j^* + \{(1 + \frac{1}{2} \hat{\eta}_4)^{1/2} - 1\} \, \frac{1}{n} \sum_{k=1}^{n} (\varepsilon_k^* - 1)]\}$$

As we show below this leads to a consistent bootstrap approximation if $c_h = 1$, i.e. if no taper is used or if the taper disappears asymptotically. We have no idea how to modify the above bootstrap for a general taper. However, in the following theorem we modify the statistic to receive a valid approximation also in the tapered case.

Franke and Härdle (1992) have used the bootstrap (2.3) without data-taper for bandwidth selection of a kernel estimate. Due to the lower rate of convergence the fourth order cumulant term disappears in the asymptotic variance for these estimates and the above problems therefore do not occur.

An estimate of $\eta_4 = \kappa_4/\sigma^4$ may be obtained e.g. by fitting a high order autoregression and calculating the empirical fourth order cumulant and the empirical variance of the estimated residuals (this is a bit contrary to the idea of a purely nonparametric bootstrap). A nonparametric estimate can be constructed in the following way (cp. Grenander and Rosenblatt, 1956, chapter 6.5): If $c_k = \operatorname{cov}(X_t, X_{t+k})$ and $d_k = \operatorname{cov}(X_t^2, X_{t+k}^2)$ then it is easy to show

$$\sum_{k=-\infty}^{\infty} d_k = 2 \sum_{k=-\infty}^{\infty} c(k)^2 + (\kappa_4/\sigma^4) \, (\int_{-\pi}^{\pi} f(\alpha) d\alpha)^2$$

i.e. we obtain with the spectral density f_2 of X_t^2

$$\kappa_4/\sigma^4 = \frac{2\pi \, f_2(0) - 4\pi \int_{-\pi}^{\pi} f(\alpha)^2 d\alpha}{[\int_{-\pi}^{\pi} f(\alpha) \, d\alpha]^2}$$

We may now obtain a consistent estimate of κ_4/σ^4 by estimating the expressions in this formula.

3. The validity of the bootstrap

To establish the validity we need the following assumptions.

(A.1) X_t, $t \in \mathbb{Z}$ is a linear process, i.e.

$$X_t = \sum_{n \in \mathbb{Z}} a_n \, \varepsilon_{t-n}$$

with i.i.d. random variables ε_t with $E\varepsilon_t^2 = \sigma^2$ and $\operatorname{cum}_4(\varepsilon_t) = \kappa_4$. Furthermore, let

$$\sum_n |a_n| < \infty \quad \text{and} \quad \inf_{\alpha \in [0,\pi]} f(\alpha) > 0 .$$

(A.2) $\hat{f}$ is a uniformly strong consistent estimate of f, $\hat{\eta}_4$ is a strongly consistent estimate of η_4.

(A.3) ϕ: $[-\pi,\pi] \rightarrow \mathbb{R}$ is of bounded variation and symmetric. Let $\phi_j \equiv \phi(\frac{2\pi j}{T})$.

(A.4) The data taper h: $\mathbb{R} \rightarrow [0,1]$ is of bounded variation with $h(x) = 0$ for $x \notin (0,1)$ and $\int_0^1 h(x)^2\, dx > 0$.

Furthermore, let $d_2(F,G) = \inf_{\substack{X\sim F\\ Y\sim G}} \{E(X-Y)^2\}^{1/2}$ be the Mallow's metric and $c_{hT} = T\, H_{4,T} / H_{2,T}^2$.

Theorem. Assume (A1) – (A5). Then we have for the bootstrap procedure (2.7)

$$d_2(\sqrt{T}(A(\phi,I_T) - A(\phi,f)),\ \sqrt{Tc_{hT}}\ (B(\phi,I^*) - B(\phi,\hat{f})) \rightarrow 0 \quad \text{a.s..}$$

Proof. We only give a sketch. It is sufficient to prove the weak convergence of both statistics to the same limit and the convergence of the second moments. For $\sqrt{T}(A(\phi,I_T) - A(\phi,f))$ this follows e.g. from Dahlhaus (1983, Theorem 2). Standard time series calculations yield for the conditional expectation and the conditional variance of $\tilde{\varepsilon}_j$ given the original sample

$$E^* \tilde{\varepsilon}_j = 1$$

and

$$\tau_T := \text{var}^* \tilde{\varepsilon}_j = \{\frac{1}{n} \Sigma_{j=1}^n (I_j/\hat{f}_j)^2 - 1\}$$
$$/ \{\frac{1}{n} \Sigma_{j=1}^n I_j/\hat{f}_j)\}^2 \rightarrow 1$$

almost surely. Direct calculations now show that

$$\text{cov}^*(I_j^*, I_k^*) = \tau_T\ \hat{f}_j\ \hat{f}_k \{\delta_{jk} + \widehat{\eta}_4\, T^{-1}\}$$

(i.e. we have emulated the expression (2.6)). This implies

$$\text{var}(\sqrt{Tc_{hT}}(B(\phi,I_T^*) - B(\phi,\hat{f}))$$
$$= c_{hT}\, \tau_T \{\frac{2\pi^2}{n} \Sigma_{j=1}^n\ \phi_j^2\ \hat{f}_j^2 + \widehat{\eta}_4\ (\frac{\pi}{n} \Sigma_{j=1}^n\ \phi_j\ \hat{f}_j)^2\}$$

which almost surely tends to v as in (2.2). Since

$$\sqrt{Tc_{hT}}(B(\phi,I_T^*) - B(\phi,\hat{f})) = \sqrt{Tc_{hT}}\, \frac{\pi}{n} \Sigma_{j=1}^n$$
$$\{\phi_j \hat{f}_j + (\hat{d}_4/\pi)\, \frac{\pi}{n} \Sigma_{k=1}^n\ \phi_k\ \hat{f}_k\}\, (\varepsilon_j^* - 1)$$

with $\hat{d}_4 = \{1\ +\ \frac{1}{2}\ \widehat{\eta}_4\}^{1/2} - 1$ the asymptotic normality follows from the central limit theorem for a triangular array of independent variables.

We therefore have found a frequency domain bootstrap which also works in the non-Gaussian case. Concerning the data taper the result is not satisfying since we emulate the increase of the variance only by a constant. However, in the case of an asymptotically vanishing taper (which is a realistic assumption from a practical point of view) we may omit the factor c_{hT} .

References

Brillinger, D.R. (1981). Time Series: Data Analysis and Theory. Holden Day, San Francisco.

Dahlhaus, R. (1983). Spectral analysis with tapered data. *J. Time Ser. Anal.* **4** 163 - 175.

Franke, J. and Härdle, W. (1992). On bootstrapping kernel spectral estimates. *Ann. Statist.* **20**, 121 - 145.

Freedman, D. A. (1984). On bootstrapping two-stage least-squares estimates in stationary linear models. *Ann. Statist.* **12**, No. 1, 827 - 842.

Grenander, U. and Rosenblatt, M. (1956). Statistical Analysis of Stationary Time Series. Almgrist and Wiksell, Stockholm.

Hurvich, C. M. and Zeger, S. L. (1988). Frequency domain bootstrap methods for time series. *Unpublished manuscript*. Department of Statistics and Operations Research, New York University

Kreiss, J. P. and Franke, J. (1992). Bootstrapping stationary ARMA-models. *J. Time Ser. Anal.*. Vol. **13**, No. 4, 297 - 317.

Künsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *Ann. Statist.* **17**, 1217 - 1241.

Liu, R. Y. and Singh, K. (1992). Moving blocks jackknife and bootstrap capture weak dependence. In Exploring the Limits of Bootstrap, ed. by LePage and Billard. Wiley, New York.

Politis, D. N. and Romano, J. P. (1992). A general resampling scheme for triangular arrays of α-mixing random variables with application to the problem of spectral density estimation, *Ann. Statist.* **20**, 1985 - 2007.

A Robust Visual Access and Analysis System for Very Large Multivariate Databases

T. Mihalisin, J. Schwegler and E. Gawlinski
Temple University, Phila. PA 19122,
and Mihalisin Associates, Inc.

J. Timlin and J. Mihalisin
Mihalisin Associates, Inc.
600 Honey Run Rd., Ambler, PA 19002

Abstract

A new system is discussed which allows one to access data buried in very large complex databases far faster, literally thousands of times faster, and with more data insight than is possible using conventional relational database management systems. The system known as TempleMVV is based on U.S. Patent No. 5228119. It allows users to visually select records based on criteria imposed on one, two or up to ten independent variables and/or on the minimum, maximum, mean, sum or standard deviation of a dependent variable in any or all subspaces of the ten dimensional independent variable space. A multidimensional graph of the data is in view during the selection process. The independent variables may be categorical, ordinal, continuous or any mixture thereof. Data involving tens of millions of records and ten variables can be viewed in seconds on a 486 computer running Microsoft Windows or a UNIX workstation.

Introduction

The technology to collect and store vast quantities of data has grown rapidly over the past decade. Satellite surveys, credit card reports and supermarket scanner records are all testaments to the perceived importance of collecting information. Unfortunately the techniques available to analyze and utilize these tremendous data warehouses are limited.

The multivariate problem in general has many difficulties, but these problems are compounded by the quantity of data involved. Slow access times make even routine tasks tedious. Interactively exploring the dataset is nearly impossible with conventional methods.

In previous papers [1-7], we have described a novel method for analyzing multivariate data, called MultiVariate Visualization In this paper we apply MVV to the problem of accessing the information in very large databases.

A Discrete Approach

MVV treats the three types of variables — categorical, ordinal and continuous — on an equal footing. Continuous variables are binned into intervals, ordinal variables may be grouped, and categorical variables are assigned an order. This converts any type of variable into a sequence of" bins". Various binnings may be chosen to change the available resolution and/or different orderings for categorical variable values may be better suited for different analyses.

The Multivariate Summary Tree

By binning the n variables, the n-dimensional space is divided into $N = b_1 * b_2 * b_3 * \ldots * b_n$ segments called primitive cells, where b_i is the number of bins for the ith variable. Each primitive cell corresponds to a unique specification of bin values for the variables.

A "multivariate summary" is created by tabulating the following five statistics for each primitive cell. The number of records in the cell are counted, and for one or more dependent variables the sum and the sum of the squares, as well as the minimum and the maximum are calculated.

After specifying an order for the variables, a tree structure is created. The primitive cells are associated with the leaves, or the bottom level, of the tree. The nodes for the next level of the tree combine the values for the first, or fastest running,

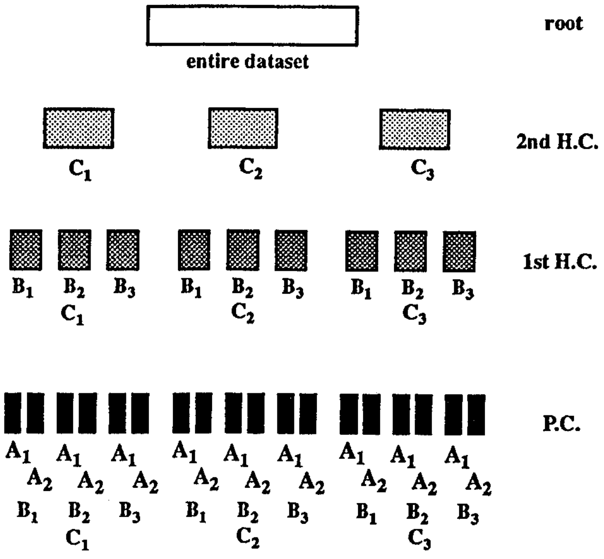


Figure 1 - The tree structure for three independent variables A, B, C with 2, 3 and 3 bins respectively. Here A is called the "fastest" variable and C is the slowest.

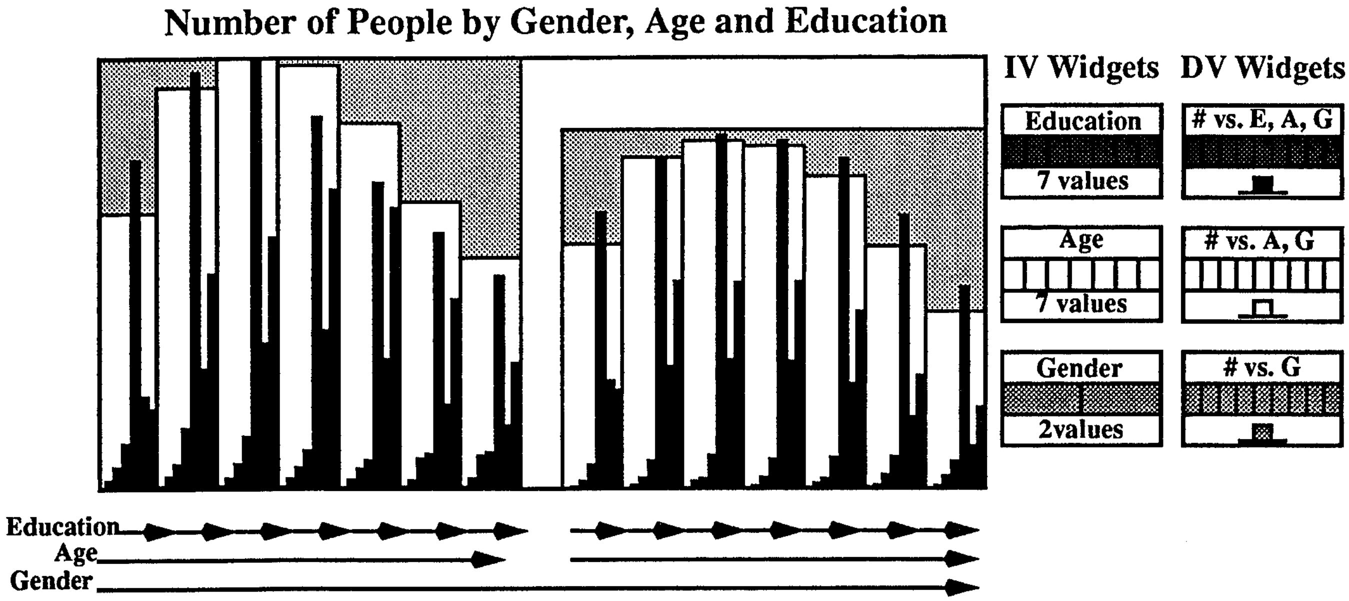


Figure 2 - The number of people N with a given level of education E in a specific age bracket A of a particular gender G is represented by the heights of the narrow black vertical bars. The number N for given A and G irrespective of E is represented by the wider white bars. Finally, N for each G irrespective of E and A is represented by the two gray bars. Each bar type has its own scale.

variable, creating statistics which correspond to specific bin values for the other n-1 variables. The next level excludes the second variable, and so on. A total of n+1 levels (including the root) are generated like this, with each level showing progressively less detail. The root of the tree contains the five statistics for the entire data set.

The resulting data structure is much more easily manipulated than the original dataset (its size depends only on the number of variables and their resolution — not on the size of the original dataset), yet it retains most of the multivariate information of the original data. If reference to the original records is required, this may be included for the relatively small overhead of a single integer per record.

Figure 1 shows the nodes of an MVV tree for the simple case of three variables A,B and C with 2,3 and 3 bins respectively. Variable A is called the "fastest running" variable because it cycles through its values A1 and A2 (at the bottom of the tree) faster than does variable B or C. B is the second fastest and C the slowest. Nodes at the bottom of the tree contain the five statistics described above for all records in each of the 2*3*3=18 "primitive cells" (P.C.). Each primitive cell corresponds to spanning just one bin for each of the three variables A, B and C. The first level of the tree above the bottom level consists of nodes which contain the five statistics for all records in the "first hierarchical cells" (1st H.C.). These cells span just one bin for variable B and one for variable C but span all bins for the fastest variable A (here just A1 and A2). Similarly the next tree level up consists of nodes which contain the statistics for "second hierarchical cells" (2nd H.C.) which span all A and B bins but only one C bin. Finally the top level of the tree has just one node which contains the five statistics for the entire dataset i.e. the "root cell" which spans all A,B and C bins.

These five statistics allow one to recursively calculate the number of records as well as the minimum, maximum, mean, standard deviation, standard deviation of the mean and sum for any variable chosen as the dependent variable of interest at all node levels of the tree. One or more of these statistics can then be used to drive attributes of symbols such as their size, location or color.

A Nested, Hierarchical Display

Figure 2 shows census data. Here N, the number of records, corresponds to the number of people and is broken down by gender, age and education. These variables have 2,7 and 7 bins respectively with education the fastest running variable and gender the slowest. The sum statistic (of number of people) is used to drive the symbol attribute, which in the case of Figure 2 is just the height of vertical bars. Each of the three bar types has its own scale.

The black bars correspond to the primitive cells where education, age and gender have each been specified. The white bars represent the number of people for a specific age and gender irrespective of education, i.e. education has been summed over. These are the "first hierarchical symbols" corresponding to the statistics (here the sum) for the first hierarchical cells. Finally, the two gray bars represent the number of people by gender alone. These are the second hierarchical symbols.

Also shown in Figure 2 are a set of "IV Widgets" and a set of "DV Widgets" The bins for the independent variables are represented symbolically by the IV widgets. The DV widget bins represent the range of values for the chosen dependent statistic (sum, mean, etc.).

Visually Guided Data Access

For a single variable, a histogram is a valuable aid for determining which intervals of the variable are of interest. For two variables, there are many distributions which cannot be deduced from their marginal histograms alone. For more than two variables, the possibilities for complex behavior increase. A truly multivariate graph can help one specify ranges of interest, pick out important features or determine overall trends.

For example, in figure 3 the size of the circles is proportional to the number of houses. We can clearly see a strong correlation between price and size for houses, moreover we can see how this correlation shifts with location.

In previous papers, we have discussed the utility of the MVV technique for performing statistical analyses. Here we focus on using the graphics to intelligently access specific portions of a large database.

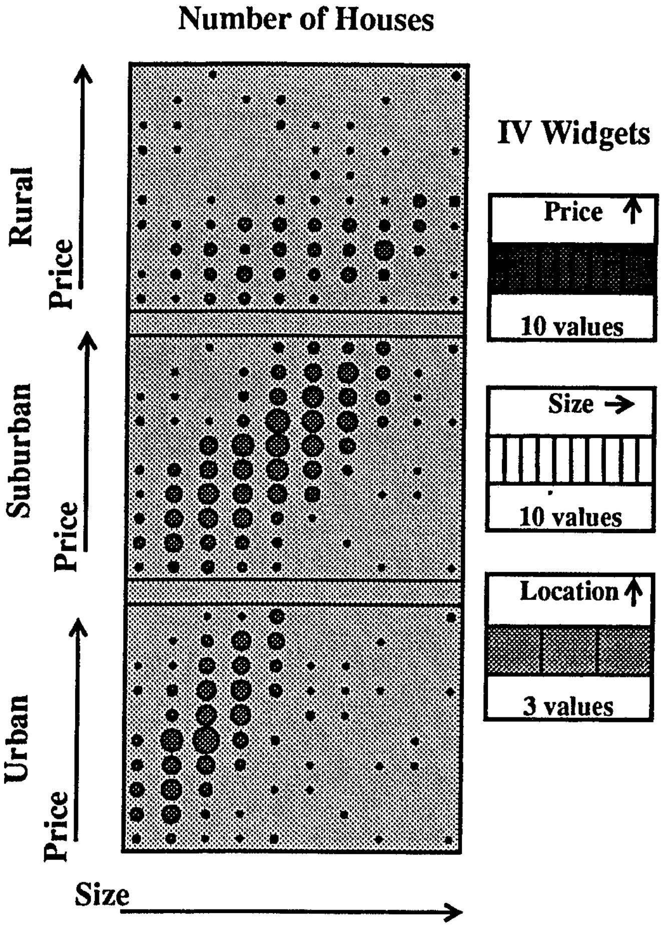


Figure 3 - A plot of the number of houses (indicated by the size of circles) versus price, size and location. Price, size and location have been binned to 10, 10 and 3 values.

IV Range Restriction

The simplest way to choose a subset of records is to restrict one of the variables to a portion of its range. We can represent this symbolically by coloring just the selected bins on the IV widget. This selects just those records whose value for the chosen variable is within the specified range. The ability to see the distribution is obviously an advantage in making these selections.

Consider figure 3. The slowest running variable is location (urban, suburban, rural). Selecting a single bin for this variable corresponds to picking all primitive cells in one of the three rectangular subgraphs. Restricting a faster running variable gives a differently shaped subset of primitive cells. Figure 4 shows some examples for the case of four variables.

Since including many variables can quickly lead to complicated graphs, an alternative is to restrict a variable which is not shown as an independent variable on the MVV graph. By adjusting the restriction, the displayed graph will evolve to display the dependence on the variable not shown. For example, we could include the age of the house as a fourth variable (represented by an IV widget) and watch how the graph of figure 3 changes as we focus on houses of different ages. While this has the advantage of keeping the graphs simple, the visual guidance for making the restriction is lost.

DV Contouring

An alternative selection method uses the dependent variable. If we think of the graph symbols as "sticking out of the page" with different sizes corresponding to different elevations, choosing all symbols of a certain size is like picking out a specific elevation on a topographic contour map. This is represented symbolically by highlighting a portion of the DV widget. Since the different symbols represent different levels of detail (depending on the number of variables included), several levels of "coarse grained" or "fine grained" contouring are available.

It is important to understand the distinction between contouring a variable as a DV versus introducing it as an IV and restricting its range. The latter (IV) restricts records on a case by case basis, the former (DV) uses the properties of a group of records. For example, the independent variable income could be used to select only those individuals in the highest income bracket. As a dependent variable, income could be used to select a group (maybe a specific age group or a specific age and education group) whose mean income has a certain value, or whose purchasing power (sum of income) is highest.

In figure 5, we show the average capital gain of stocks as a function of four indices. Contouring on the highest value of the capital gain gives the groups (as determined by the values of the four indices) with the best average performance.

DV Symbol Selection

This selection method is more robust than the previous one. Instead of choosing all symbols of a specific size, one can choose individual symbols. This is more appropriate when the relevant quantity is not the absolute value of the dependent variable but rather its value with respect to neighbors.

Consider figure 5. If we were just interested in the global maximum (or minimum) we could use DV contouring and select an extreme of the range. The graph, however, can also show us the behavior around the extrema. Both I and II indicate cells which are local maxima. However, I is a maximum which is stable with respect to changes in all four variables, but II is very sensitive to the value of C.

Depending on the dependent variable rule or statistic chosen, such a selection process could choose particularly volatile stocks, or ones which are undervalued, etc.

Other Graphical Access Systems

Johnson and Shneiderman [8] have discussed "Tree Maps" an alternative multivariate graphical data access technique. Recently Tweedie et al [9] have proposed a "drill down" type of data access tool in which parallel axes akin to those introduced by Inselberg[10] are used to display multiple (constrained or unconstrained) marginal distributions for all variables of interest. Unlike Inselberg, these authors deal only with discrete variables. That is, they follow the MVV model of binning any and all continuous variables to form discrete ones.

Although the MVV technique, Tree Maps, and the modified Inselberg approach of Tweedie et al., all utilize a multivariate graphical approach to data access they are fundamentally different in terms of their computational engines, the nature of their graphical presentation of multi-

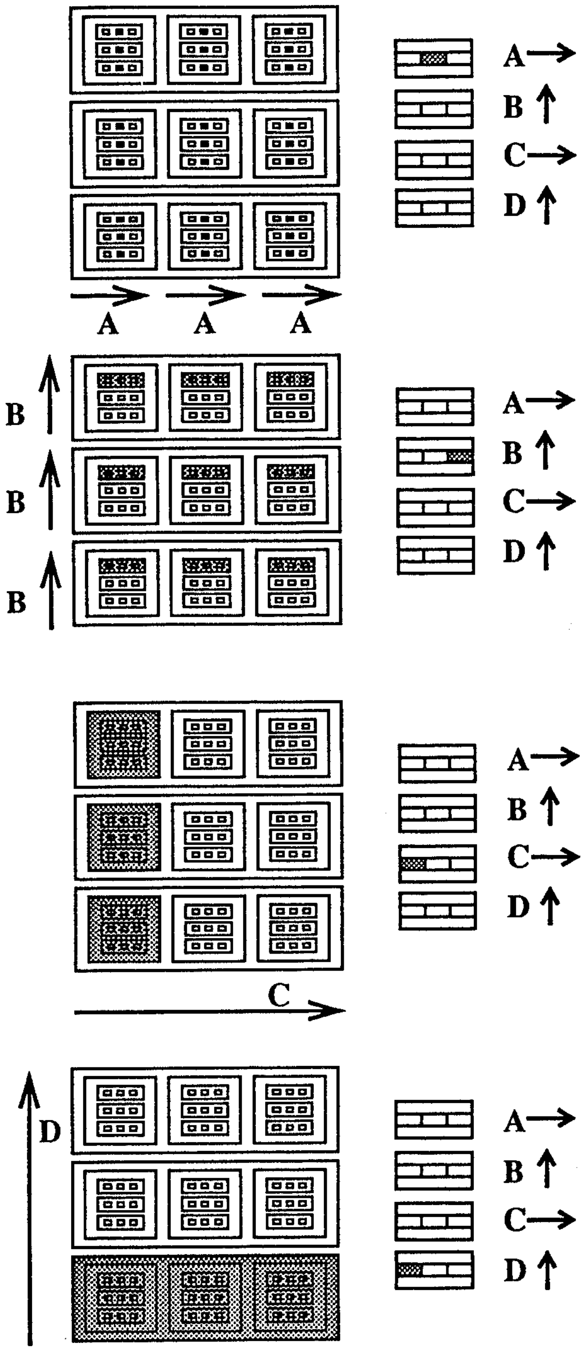


Figure 4 - Shown in parts a, b, c and d are the primitive cells that are selected when one constrains each variable (A,B, C and D respectively) to one of its three bins. Here A is the fastest and horizontal. B is the 2nd and vertical. C is 3rd and horizontal and D is 4th and vertical. In each case 27 primitive cells are selected. Constraining two IV's e.g. A as in part a and B as in part b would select the 9 primitive cells common to a and b, i.e. an intersection.

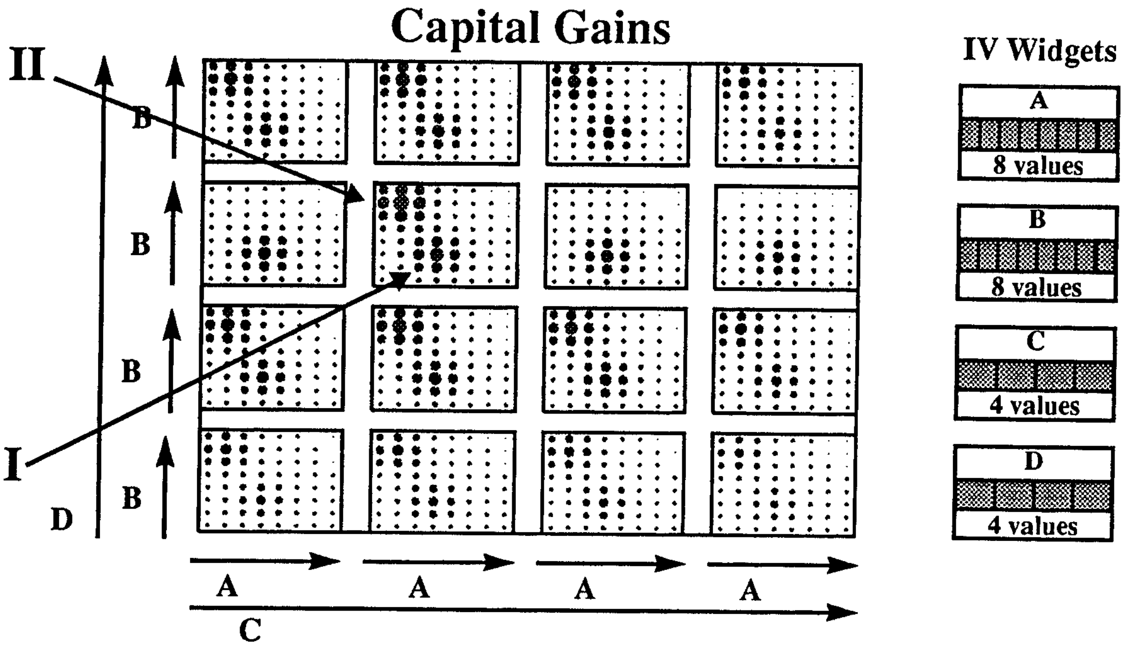


Figure 5 - The average capital gains for stocks in a four dimensional space of fundamental ratios A,B,C,D.

variate data and their scope of data analysis capabilities. The MVV method can be used to analyze and select information from very large databases consisting of literally tens or even hundreds of millions of records with subsecond response. MVV's graphical presentation can be used to find trends and correlations which may be important factors in record selection and can be generalized to displaying multiple dependent as well as independent variables.

Conclusions

By forcing all types of variables to be discrete, tremendous advantages can be gained in the manipulation of large, multivariate databases. The summary tree described above can effectively capture most of the multivariate nature of a large dataset. The nested, hierarchical graph not only displays multivariate information, but can also provide an intuitive data access system.

MVV provides one method for intelligently extracting the information from large multivariate databases.

References

1. Mihalisin, T., Gawlinski, E., Timlin, J. and Schwegler, J., Scientific Computing and Automation Vol. 6, No. 1, Oct. 1989, pp.15-20.

2. Mihalisin, T., Suntech J., Vol. 3, No. 1, winter 1990, pp.25-31.

3. Mihalisin, T., Gawlinski, E., Timlin, J., and Schwegler, J., Proc. IEEE Conf. Visualization, San Francisco, Oct. 1990, pp. 255-262.

4. Mihalisin, T., Timlin, J., and Schwegler, J., IEEE Computer Graphics and Applications, Vol. 11, No. 3, May 1991, pp. 28-35.

5. Mihalisin, T., Timlin, J., and Schwegler, J., Proc. IEEE Conf. Visualization, San Diego, Oct 22-25, 1991, pp. 171-178.

6.) Mihalisin, T., Schwegler, J. and Timlin, J., Proc. 24th Symposium on the Interface, College Station, March 18-21, 1992, ed. H. Joseph Newton, Interface Foundation of America, Fairfax Station, VA pp. 141-149.

7.) Mihalisin T., Schwegler, J. and Timlin, J., 1992 Proc. of the Section on Statistical Graphics, Boston, Aug. 9-13, 1992, American Statistical Association, Alexandria, VA pp. 69-74.

8.) Brian Johnson and Ben Shneiderman, Proc. of Visualization 91, Oct 22-25, 1991, San Diego, pp. 284 - 291.

9.) Lisa Tweedie, Bob Spence, David Williams and Ravinder Bhogal, Proc. of CHI 94, April 24-28, 1994, Boston, pp. 435-436.

10.) Alfred Inselberg and Bernard Dimsdale, Proc. of Visualization 90, Oct. 23-26, 1990, San Francisco, pp. 361-378.

Dynamic Graphics in a GIS: A Link between ARC/INFO^{TM} and XGobi

Jürgen Symanzik[1], James Majure[2], Dianne Cook[1], Noel Cressie[1]

[1] Department of Statistics, Iowa State University, Ames, IA 50011, USA

[2] GIS Support and Research Facility, Iowa State University

symanzik@iastate.edu

Abstract. This paper describes a link between a Geographical Information System (GIS), ARC/INFO^{TM} , and an interactive dynamic graphics program, XGobi. GISs provide a user with a standard and convenient software for spatial geographical data. In particular, the GIS ARC/INFO is a combination of two systems: ARC maintains the spatial information of map features and provides tools for spatial analyses while INFO maintains the thematic or attribute information associated with the map features. XGobi is an interactive dynamic graphics program for data visualization in the X Window System^{TM}. It is designed for the exploration of multivariate data, primarily by manipulating and displaying scatterplots in arbitrary dimensions.

The motivation for the work is to link the dynamic, interactive strengths of XGobi for visualizing high-dimensional data with the exhaustive map handling tools of ARC/INFO, specifically to explore spatial data. This paper presents information about the technical realization of the link between ARC/INFO and XGobi as well as an introductory example of its use.

1 Introduction

Interactive and dynamic graphics for high-dimensional data have proved useful for exploring relationships among multiple variables. Incorporating similar tools in the context of spatial data promises to be a valuable aid in exploring spatial dependencies. Geographical Information Systems (GISs) have developed sophisticated capabilities for managing multivariate spatial data bases but limited capabilities for conducting interactive exploratory data analysis. The combination of a GIS and a dynamic graphics system for multivariate data comprises a potentially powerful tool for interactive exploratory spatial data analysis.

In Section 2 we describe general features that should be available for an interactive dynamic graphics tool that operates on data available in a GIS data base. As one particular application, the interface between the GIS ARC/INFO^{TM} and XGobi, an interactive dynamic graphics program for data visualization in the X Window System^{TM}, is described in Section 3. An example is given in Section 4. We conclude this paper by describing possibilities for future work.

2 Integration of Interactive and Dynamic Graphics Tools into a GIS

The inclusion of spatial location in data analyses can be addressed in different ways. Using a GIS, we maintain a geographic context of spatial location relative to land-cover, streams, roads, and other relevant information. It would also be feasible to include the spatial coordinates as two (or d) additional variables into the analysis, but this approach by itself does not exploit the considerable spatial capabilities of GISs.

Emphasis in GIS development has been on the input of data, its management (storage, retrieval), and the display of maps, graphs, and tables. GISs have some capability to allow statistical analyses but it is generally limited. A number of recent suggestions have been made (e. g., Openshaw, 1991; Anselin and Getis, 1992; Ding and Fotheringham, 1992; Fotheringham and Rogerson, 1993) to redress this imbalance. Still others have incorporated some dynamic graphical tools into systems that lack the full features and flexibility of a GIS (e. g., Haslett et al., 1991).

Our research addresses the extremely important problem of multivariate exploratory spatial data analysis in a GIS. GIS data structures allow the representation of areal features (e. g., for the storage of information reported at an aggregated spatial level, such as counties or census tracts), linear features (e. g., for the storage of information collected from a stream or a transportation network), and point features. The topological data structure of a GIS makes it possible to determine spatial relationships between sampling locations, such as stream sites, that would be difficult to determine otherwise. The display capabilities of a GIS allow the spatial variables to be overlaid on a background of hydrography, transportation, population, land use, or other information relevant

[TM] *ARC/INFO* is a trademark of Environmental Systems Research Institute, Inc.

[TM] *X Window System* is a trademark of MIT.

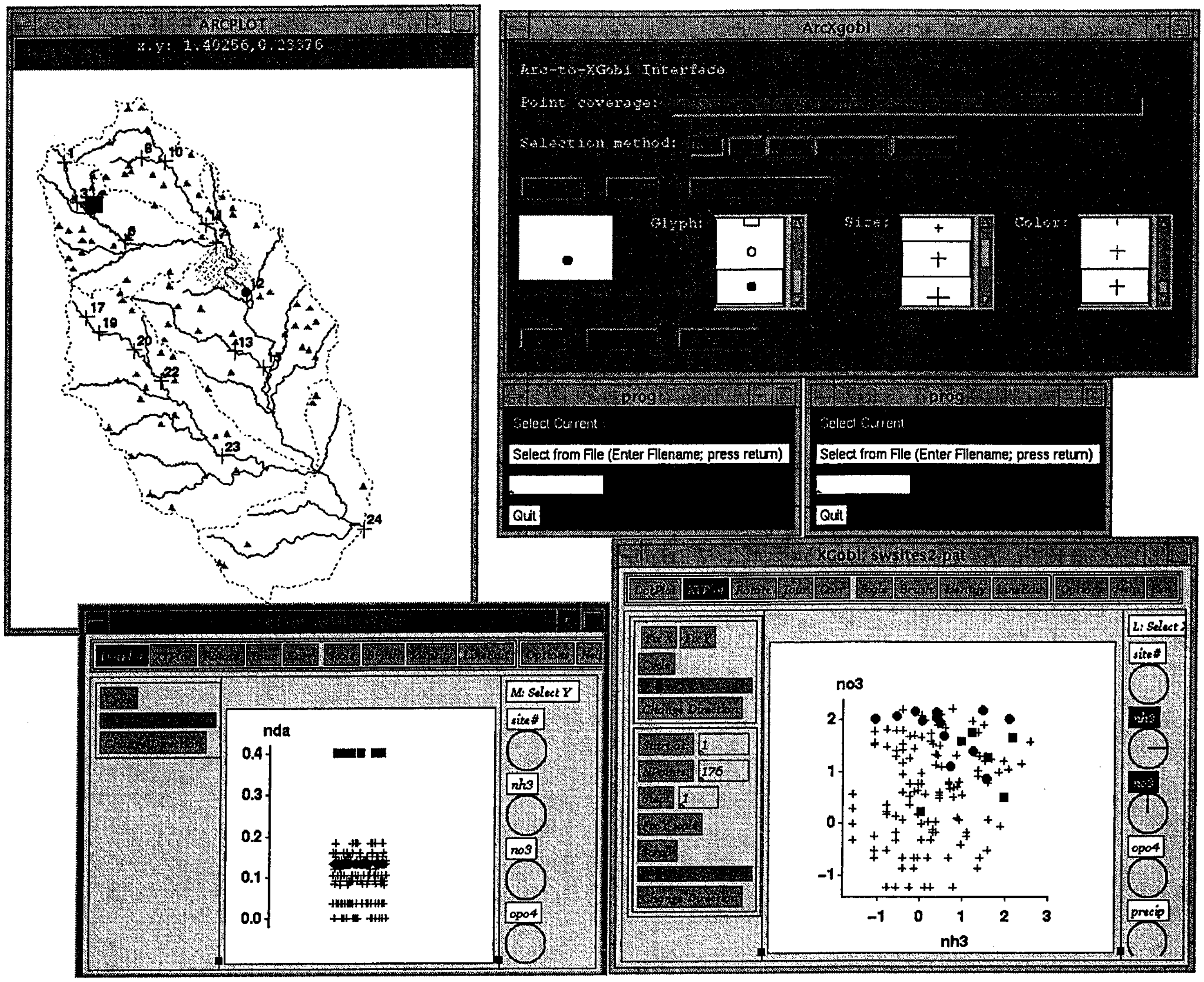


Figure 1: *ARC/INFO control panel and example map view linked to two XGobi views.*

to the attributes[1] being considered. For example, in Figure 1 the map view shows sampling sites along streams in Erath County, Texas. Information about the topography or land use near a sample site can give valuable insights into the values of attributes (e. g., ammonia concentration) collected at the site.

A GIS is intrinsically multivariate and yet this is ignored by the largely univariate statistical analyses currently available. By building an interface between a GIS and software for dynamic graphics, we will also provide a platform for developing new spatial graphical methods for spatial data sets available in the GIS (e. g., Cook et al., 1994).

3 The ARC/INFO to XGobi Interface

Our efforts have focused on interfacing the GIS software ARC/INFO with XGobi (Swayne et al., 1991). ARC/INFO has been chosen because it is one of the most frequently used GIS systems and because it is extensible through its macro language, allowing the development of menus and programs to carry out ARC/INFO tasks. XGobi provides interactive and dynamic graphical tools in the X Window System environment for exploring multivariate data through the manipulation of scatterplots. ARC/INFO is used to maintain the GIS data base and to display the geography, while XGobi is used primarily to explore the relationships within and between the attributes. Figure 2 shows how the communication between these two programs is established.

[1] In the context of GISs the expression *attribute* is used instead of the statistical expression *variable.*

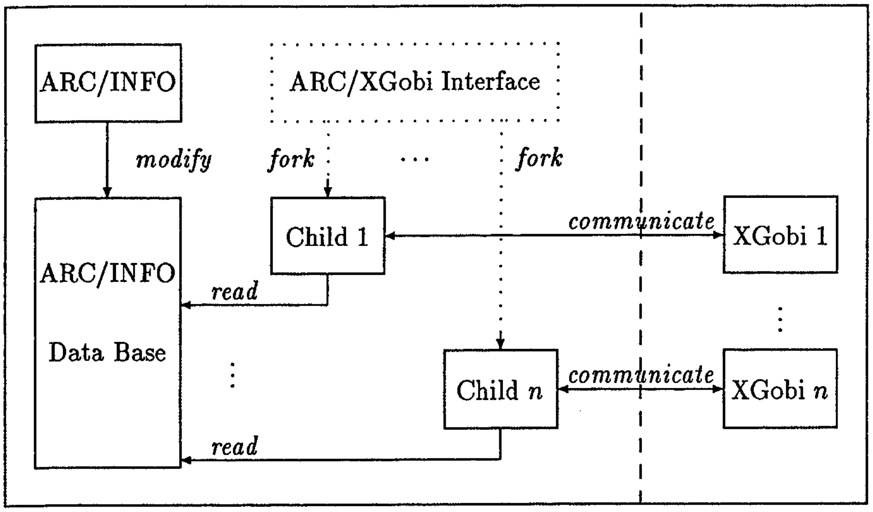


Figure 2: *Interface linking ARC/INFO with XGobi.*

3.1 The ARC/INFO Part

ARC/INFO is used to display the location of sampling sites in a graphics window. The sampling sites can be displayed on a background of roads, streams, or any other relevant geographic data sets available. A control panel, the upper right window shown in Figure 1, allows the user to brush or subset the sampling sites interactively. The term "brush" refers to changing the symbol used to represent the specified points and "subset" refers to choosing a subset of the points for further analysis, disregarding (temporarily) the other points. The brushed (or subsetted) sites are redrawn with the specified glyph, size, and color, and the ARC/INFO data base is modified. These changes are detected and passed to XGobi by the intermediate process, as described in subsequent sections.

The ARC/INFO portion of the application is implemented with AML (Arc Macro Language) and works as follows. An ARC/INFO data set consists of a set of spatial features, in this case points, each of which has a record in a data base table. When the application starts, a column in the table is initialized with a default value which represents the symbol, i. e., glyph, size, and color, with which to draw each point. As the user interactively queries the points, the values in this column are updated to reflect the user's actions. The changes to this column are detected by the intermediate ARC/XGobi server process and sent to XGobi.

Pseudo code for the ARC/INFO part is given below. In this pseudo code, the control panel is represented by the repeat loop.

```
set current symbol to default symbol
repeat
  wait for user action
  case user action {
    when "identify ARC/INFO data set"
      initialize the symbol column to current symbol
    when "brush"
      spatially select points to brush
      set symbol column of selected points to current symbol
    when "subset"
      spatially select points to subset
      set symbol column of selected points to current symbol
      set symbol column of other points to 0
    when "clear selection"
      reset the symbol column of all points to default symbol
    when "change color"
      reset current symbol to reflect changed color
    when "change glyph"
      reset current symbol to reflect changed glyph
    when "change size"
      reset current symbol to reflect changed size
  }
until (forever)
```

3.2 The Intermediate Process

The intermediate process, denoted as ARC/XGobi Interface in Figure 2, has to serve the requests of the XGobi clients by reading information from the ARC/INFO data base. The interprocess communication between this server and the XGobi clients is based on Stevens' (1990) concurrent server example, and uses a Transmission Control Protocol (TCP) socket, i. e., an Internet stream socket. Upon receiving a connection request from an XGobi client, the intermediate process forks an identical child process. Each child process communicates with one XGobi client; thus, one–to–one connections between server processes and XGobi clients are established.

Obviously, the forking of child processes is a heavy weight mechanism to provide a concurrent server. However, we assume that this mechanism is available for all hardware environments that support ARC/INFO. An alternative for some workstations (e. g., DEC$^{\text{TM}}$) is the use of multithreads which are light weight processes, but this approach is not available on all systems (e. g., Sun$^{\text{TM}}$/Sparc$^{\text{TM}}$ workstations).

The main task of the child processes is the following: If the XGobi client indicates that it wants the currently selected ARC/INFO data set and future updates of this selection, the related ARC/XGobi server (child) has to check continually whether the ARC/INFO data base has been changed. If so, the modifications, such as new brushed or subsetted points, are immediately passed to the corresponding XGobi clients.

A child process is terminated by a QUIT command of its XGobi counterpart, or if it detects the unexpected termination of the client process or the breakdown of the communication channel. The intermediate (parent) process will operate until it is explicitly terminated by the user. The pseudo code for the ARC/XGobi interface follows.

Parent:

```
init ARC/INFO defaults
init sockets
repeat
  accept connection from XGobi client
  fork child process
until (forever)
```

Children:

```
repeat
  wait for input from XGobi client or for Timeout
  if (input received = SEND Filename)
     then {send data from file Filename; Update = false}
  else if (input received = SEND current)
     then {send data from current selection; Update = true}
  else if (Timeout and Update)
     then if (current selection modified since last send)
             then send update of current selection
until (input received = QUIT or abort of client
                               or channel down)
```

3.3 The XGobi Part

There are several methods that we considered when initially contemplating a link from ARC/INFO to XGobi: directly writing new functionality into XGobi, accessing the XGobi data structures by calling XGobi as a subroutine, or using the linked brushing protocols existing in XGobi. The first method is feasible because the code for XGobi is available, but it is undesirable because it would require maintaining updates with new releases of the XGobi code. The third option strictly limits the interaction to the data structures available in the XGobi linked brushing code. Calling XGobi as a subroutine from a small control panel was chosen as the method that best suited our needs. Almost all the data structures used in XGobi are available for modification using this approach.

The structure of the calling program is based on the subroutine template code provided with the XGobi source code. (The subroutine approach also has been used by Littman et al., 1992, for the implementation of the XGvis software system.) A control panel is initiated for each instance of XGobi (see Figure 1), from which the user has the option of selecting data from an ARC/INFO data base file or to receive the data set that is currently selected within ARC/INFO. Once the data source has been determined and the data received, the XGobi window is initialized.

Internally, an additional working procedure, namely a routine that runs once whenever the X Window System event loop finds no events, has been added to XGobi to check for incoming data from the ARC/XGobi server. If this routine receives updates of the data, the attribute values currently visible in XGobi linked to the brushed or subsetted coordinates in ARC/INFO will instantaneously be set to the same glyph, size, and color.

Otherwise, the entire functionality of XGobi has been maintained. The XGobi part can be described via the following pseudo code.

```
init sockets
connect to ARC/XGobi server
init XGobi defaults
init startup window
repeat
  wait for user input
  send input to ARC/XGobi server
  wait for data from ARC/XGobi server
  if (XGobi not invoked)
     then invoke XGobi
     else update XGobi structures and data sets
until (user input = QUIT)
```

3.4 Usage

ARC/INFO and the ARC/XGobi (parent) interface process must be activated on the host where the ARC/INFO data base is located. Then, XGobi client processes can connect to the ARC/XGobi server. Clients can reside on the same host or anywhere else on the Internet. Internet addresses, ports, and communication protocols are encoded into the program. So, the user does not have to worry about common setups for server and clients. If the user only wants to use XGobi to analyze the attribute data in an ARC/INFO data set, the invocation of ARC/INFO is not required.

$^{\text{TM}}$*DEC* is a trademark of Digital Equipment Corporation.
$^{\text{TM}}$*Sun* is a trademark of Sun Microsystems, Inc.
$^{\text{TM}}$*Sparc* is a trademark of Sun Microsystems, Inc.

4 An Example

As an example of how the link between ARC/INFO and XGobi can be used to explore data we show a data set containing water–quality data collected during several weeks at seventeen surface–water sampling sites in Erath County, Texas (see Figure 1). The pollutants are being modelled inter alia through explanatory variables, such as the number of dairies per acre or the number of head of cattle per acre, to account for large–scale variability.

As well as the sampling sites (numbered from 1 to 24 with some numbers missing), the ARC/INFO mapview shows streams (continuous lines), boundaries of large basins (dashed lines), dairies (triangles), and a town (shaded area). After an initial examination of the map, two of the sampling sites, numbered 4 and 12, have been brushed in order to see if the data collected there is anomalous.

Site 4 has been brushed because it is at the outlet of a very small basin containing four dairies. Thus, the response variables (i. e., pollutants) might be expected to be unusually high. The XGobi view on the left shows that the explanatory variable "nda" (the number of dairies per acre) is extremely large relative to the other sites. The XGobi view on the right shows that the responses "no3" (standardized nitrate) and "nh3" (standardized ammonia) at this site are high, though not outlying.

Site 12 has been brushed because it is located just below the town of Stephenville, Texas; the waste water treatment plant of Stephenville discharges into the stream above the sampling site. The XGobi view on the right shows that standardized nitrate is consistently high at this site, but nothing remarkable can be said about standardized ammonia. Based on this result, an analyst doing an exploratory data analysis probably would be interested in how strong the nitrate concentration is further downstream of Stephenville and, therefore, the next site to be brushed in the ARC/INFO view might be site 24.

This example demonstrates briefly how geography can give an analyst insight into the exploration of a data set and, thus, why a link between ARC/INFO and XGobi is a useful tool.

5 Future Work

We have presented an interface linking ARC/INFO with XGobi. This interface allows the user to link ARC/INFO and XGobi views interactively such that modifications of the ARC/INFO view automatically change the other views in the different XGobi clients. So far, this link is only unidirectional. Future work will focus on the other direction of the link, that is, the update of the ARC/INFO view according to one (or several) XGobi view(s). However, this direction is more problematic since substantial questions, such as security (Who is allowed to modify the data?), concurrency (What if two XGobi clients send different update information at the same time?), and technical issues (How can events be incorporated into a primarily non–event–driven FORTRAN program?) have to be resolved. We are also looking towards a new release of ArcView^{TM} . According to preliminary announcements from the distributors of this software, this new version seems to be more suitable to facilitate the inverse link from XGobi to ARC/INFO.

Acknowledgements

Symanzik's research was partially supported by a German "DAAD–Doktorandenstipendium aus Mitteln des zweiten Hochschulsonderprogramms". Cressie's research was supported by the National Science Foundation, the National Security Agency, and the Office of Naval Research. We thank Deborah Swayne for her valuable assistance in integrating XGobi into our application.

References

Anselin, L. and Getis, A. (1992). Spatial Statistical Analysis and Geographic Information Systems. *Annals of Regional Science*, 26:19–33.

Cook, D., Cressie, N., Majure, J., and Symanzik, J. (1994). Some Dynamic Graphics for Spatial Data (with Multiple Attributes) in a GIS. In *COMPSTAT '94–Proceedings (To appear)*.

Ding, Y. and Fotheringham, A. S. (1992). The Integration of Spatial Analysis and GIS. *Computers, Environment and Urban Systems*, 16:3–19.

Fotheringham, A. S. and Rogerson, P. (1993). GIS and Spatial Analytical Problems. *International Journal of Geographical Information Systems*, 7:3–19.

Haslett, J., Bradley, R., Craig, P., Unwin, A., and Wills, G. (1991). Dynamic Graphics for Exploring Spatial Data with Application to Locating Global and Local Anomalies. *The American Statistician*, 45(3):234–242.

Littman, M., Swayne, D. F., Dean, N., and Buja, A. (1992). Visualizing the Embedding of Objects in Euclidean Space. *Computing Science and Statistics*, 24:208–217.

Openshaw, S. (1991). Developing Appropriate Spatial Analysis Methods for GIS. In Maguire, D. J., Goodchild, M. F., and Hind, D. W., editors, *Geographical Information Systems: Principles and Applications, vol. 1*, pages 389–402, London. Longman.

Stevens, W. R. (1990). *UNIX Network Programming*. Prentice–Hall, Englewood Cliffs, NJ.

Swayne, D. F., Cook, D., and Buja, A. (1991). XGobi: Interactive Dynamic Graphics in the X Window System with a Link to S. In *ASA Proceedings of the Section on Statistical Graphics*, pages 1–8, Alexandria, VA. American Statistical Association.

^{TM}*ArcView* is a trademark of Environmental Systems Research Institute, Inc.

Variations on Row-Labeled Plots For Reexpressing Tabular Summaries[1]

By

Daniel B. Carr and Kwang-Su Yang
George Mason University
Fairfax, VA 22030

Abstract

This paper introduces the task of converting summary tables into row-labeled plots. The conversion task emphasizes exposition of important patterns in data rather than data archival. Attention to graphical design details yields plots that appear simple even though the tables are fairly complex. The more general task includes converting multiway tables and distributional summaries to plots. This brief paper focuses attention on two templates for expressing two-way tables as plots. These templates are variations on familiar dot and bar plots and have numerous applications.

1. Introduction

This paper advocates the use of row-labeled plots for graphical presentation of tabular information. Row-labeled plots (or row plots for short) take three basic forms, dot plots (charts), horizontal bar plots (charts), and horizontal distributional summary plots, such as boxplots. While the plots are familiar, government reports still seem to favor tables over plots. Numerous reasons can be cited for the common usage of tables: historical inertia, an emphasis on data archival rather than on communication, limited access to software that produces presentation quality graphics (especially for dot plots), and an absence of graphical paradigms for handling the challenges posed by reexpressing tabular information. As part of the advocacy for row plots, this paper provides software and paradigms that address several of these challenges.

The basic tasks in converting tables to plots involve accommodating the tabular structure and emphasizing chosen comparisons. Structure related challenges include representing several factors, handling nested factors, showing many levels within a factor, providing resolution for a large range of values, and showing distributional summaries. Emphasis considerations include stressing estimates over confidence bounds and calling attention to the more accurate estimates. Carr (1994) provides examples for all of these cases including redesigned boxplots. This paper presents two templates for converting two factor tables to plots.

2. Two-Factor Row Plots

In row plots the levels of one factor become rows. Row plots accommodate a second factor in one of three ways: by using symbols, by using multiple panels, or by showing the levels of both factors as rows. Figure 1 provides and example using symbols. Rows represent the 16 levels of the carcinogens factor. Symbols represent the two levels of the years factor. Representing the two levels of the second factor using symbols is advantageous. All the values can be compared using a single common scale. No space is lost through adding panels or rows.

The symbols used in Figure 1 emphasize change. Open circles show the 1987 values and the arrow tips designate the 1988 values. A horizontal line from the 1987 value to the 1988 value explicitly shows the change. When the change is small, a circle with a dot represents both the 1987 and the 1988 values. This reflects a willingness to make small adjustments in symbol placement (the 1988 value) and style for graphical simplicity and clarity.

Several additional facets of Figure 1 reflect design considerations: grid lines, sorting and grouping of rows, and the log scale. The horizontal grid lines in Figure 1 are white lines on a light gray background. The gray background gives the plot a value-added appearance. The small contrast between the light gray and white lines allows the lines to be perceived as part of the background rather than competing with the symbols in the foreground. The horizontal lines help in table lookup (matching of labels and symbols).

Two design considerations, sorting and grouping of rows help the plot appear less complex. The sorting of rows by the 1987 values reduces the visual distance between the symbols as the reader scans the plot vertically. The conjecture here is that reducing the visual distance in looking from point to point makes the plot appear less complex.

[1] This work was supported by EPA under cooperative agreement No. CR8280820-01-0. The article has not been subject to peer review of the EPA and thus does not necessarily reflect the view of the agency and no official endorsement should be inferred.

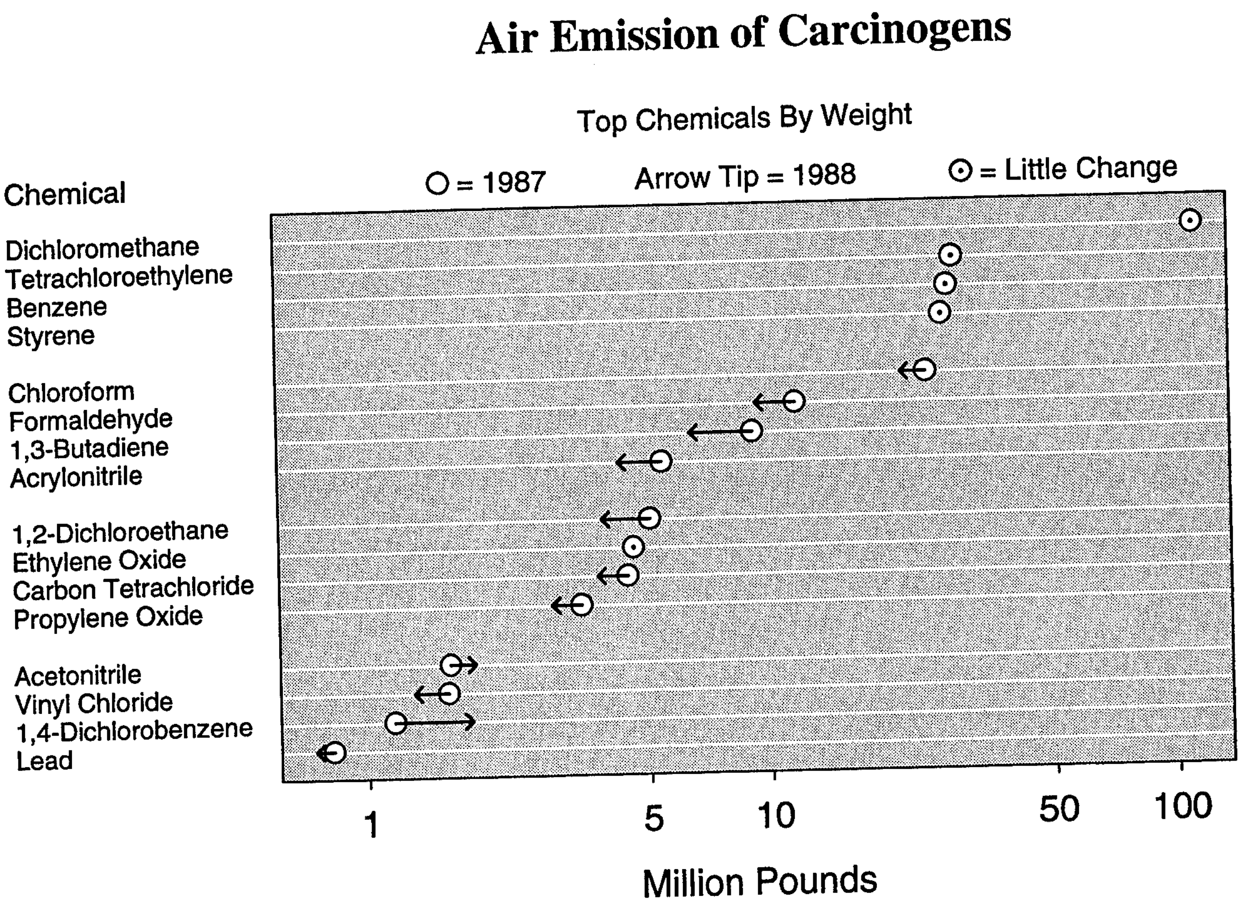


Figure 1. A row plot with symbols representing the levels of a second factor.

The grouping of rows in Figure 1 creates smaller perception units. Four groups of four appears more manageable than one group of sixteen. Grouping rows also facilitates the matching of symbols to row labels. While the horizontal grid lines help, grid lines are not so crucial with grouping because matching say the third of four labels with the third of four symbols is trivial. The grouping of rows diminishes the advantage of using right-aligned row labels. Cleveland's examples (1984, 1985, 1993a, and 1993b) show the evolution from left-aligned labels to right-aligned labels. Right-aligned labels are closer to the horizontal grid lines and corresponding symbols, so right-alignment should reduce the chances of making an error in matching labels with symbols. However, the conjecture here is that grouping makes the error rate very low so that there is little advantage in using right-aligned labels. Here the preference is to follow the conventions for the dominant activity in each part of the plot. Reading is the dominant activity in the row-label part of the plot so the labels are left-aligned.

Figure 1 uses a log scale. The carcinogens selected for the table motivating Figure 1 represent the most extreme cases in terms of pounds. The range of values for extreme cases is often large and using a log scale helps to provide resolution for the smaller values. The difference of values on a log scale is a monotonic function of the percentage change. The log scale is fine for mathematically sophisticated audiences. For more general audiences, a plot showing percentage change on a linear scale would be helpful. Of course sorting rows by percentage change and the other design

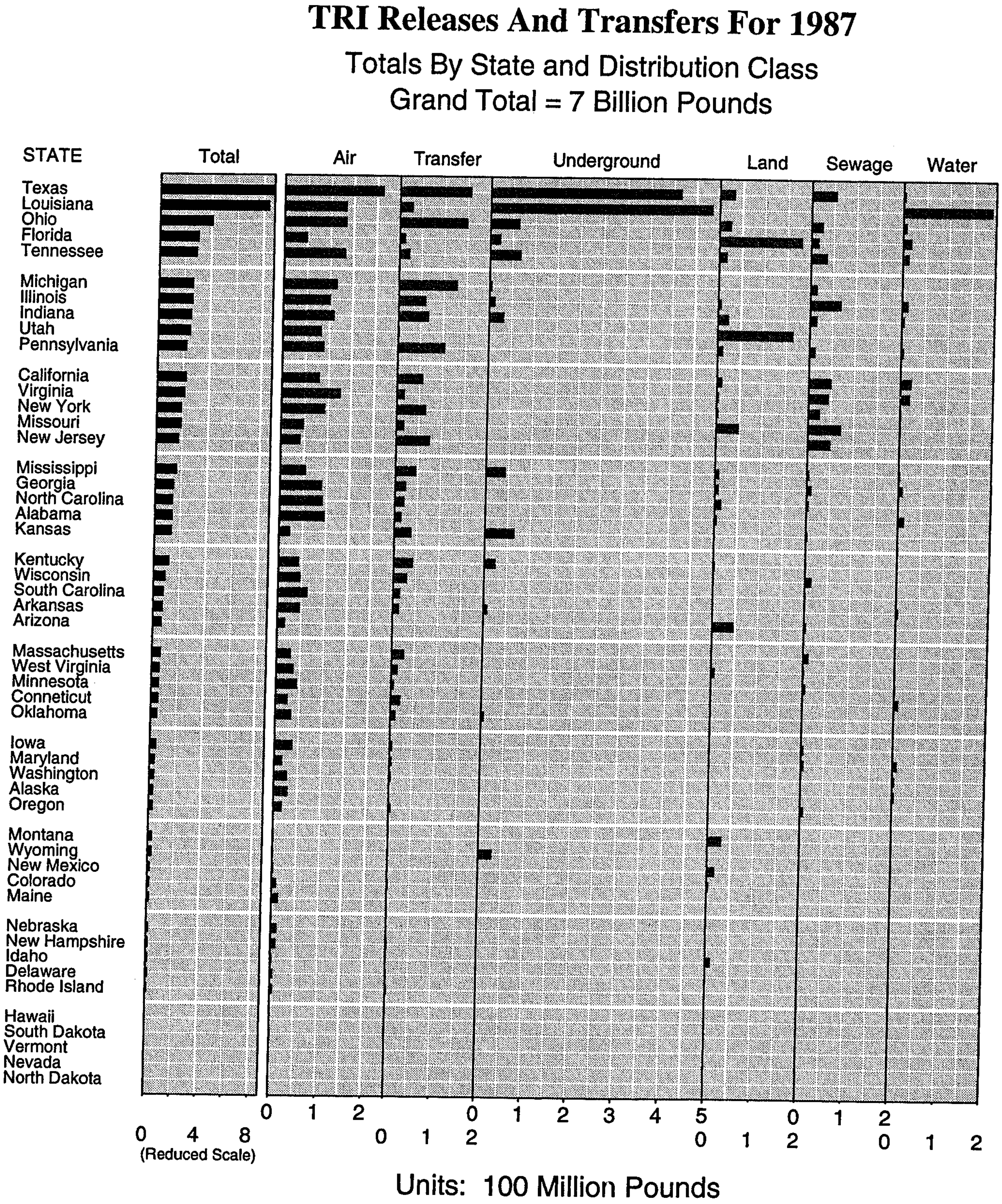


Figure 2. A row plot with panels representing the levels of a second factor.

considerations still apply.

Sorting and grouping considerations carry over to multiple panel displays as in Figure 2. Figure 2 uses rows to distinguish the values for the 50 states. In this case the design groups rows in units of five. Grouping rows into units of four likely has some cognitive advantages over units of five, but producing ten full groups of five seems reasonable. The toxic release class factor has 6 levels plus a marginal summary. Representing seven levels using symbols is a bad idea. In fact Kosslyn (1994) suggests that distinguishing among more than four elements gets complicated. Consequently Figure 2 represents the levels of the second factor using multiple panels rather than symbols.

Figure 2 reflects several additional design considerations. The plotted symbols are bars. Bars are visually dominant area symbols that allow the reader to quickly scan the whole plot even though there are separating panel lines. Note that the bars in the right-most six panels are on the same scale so are directly comparable.

The common approach to creating comparable bars is to use identical width panels with identical scales that cover the full range of data. The result of this approach is that panels with small values are largely blank. The right-most six panels in the Figure 2 have different widths. Since the "underground" values are the largest the "underground" panel is widest. This unequal panel width approach makes effective use of the available space while preserving comparability. Given a fixed plotting space, the range-driven panel width approach uses the otherwise blank space to increase the resolution within all panels.

Figure 2 includes vertical grid lines. Cleveland (1993a, 1993b) provides a demonstration that shows how helpful grid lines are in making more accurate comparisons across panels. Cleveland notes that the grid lines allow attention to be focused on smaller graphical elements and that Weber's law helps to explain the increased accuracy of comparison. Grid lines are an important facet of the graphical design.

Figure 2 uses a different scale for state totals panel to save space for the other panels. The figure calls out this different scale in four ways, by using black bars rather than dark gray bars as in the other panels, by the slight separation from the other panels, by warning text below the panel and by the tic labels. In addition, the grid spacing turns out to be different. The design places the state total panel appear first among the panels because it is an executive summary and the basis for the sorting of rows.

Figure 2 provides a quick state-based overview of the toxic releases for the different release classes. The table motivating this plot appears in Courteau 1990. The table spans pages 37 and 38 and the summarizes the company self-reports with nine digits of accuracy. The table is truly a visually intimidating table. Table design considerations such as rounding numbers, sorting rows and grouping rows can substantially improve the table for exposition purposes. However most readers will still prefer the graphical summaries like Figure 2.

A single plot will not necessarily cover all the major exposition objectives for a table. Most people are interested in the values for their state. People from states like Hawaii, won't see much in Figure 2. More resolution is desirable. An additional plot showing the six release class values as percentage of state totals would be helpful as a summary. Individual state maps can show further detail. The tabular summaries of the Toxic Release Inventory can lead to many visual representations.

3. Comments and Conclusions

Historically, tables in government documents served a data archival role. Today electronic storage better serves this archival role. Government publications need to change from an archival orientation to a data exposition orientation. While some tables may remain because tables can be advantageous for careful quantitative analysis, most people prefer the more qualitative visual understanding provided by plots. The development of graphics templates and corresponding software will facilitate making this change.

The government community primary has access to the spreadsheet-based business graphics developed in the 1970's. Much has been learned about graphic design in the last two decades. Those that study graphic design know that stacked bar plots and pie charts are inferior visual representations of data. Nonetheless such graphics commonly appear in government publications because the available software make the graphics easy to produce.

The two figures in this paper provide templates for converting commonly encountered two-factor tables in to plots. More cases are covered in Carr (1994). The examples have face validity. If they appear better than numerous alternatives then likely they are better. However the examples have not been subject to rigorous cognitive tests and the plot that cannot be improved is exceedingly rare. Readers are welcome to develop their own templates for converting table to plots. Those wanting to modify or use the current templates can build upon the current work. The data, Splus functions and script files are in /pub/submissions/rowplot on galaxy.gmu.edu and can be obtained by anonymous ftp.

References

Carr, D. B. (1994). "Converting Tables to Row-Labeled Plots," *Technical Report No. 101*, Center for Computational Statistics, George Mason University, Fairfax, VA.

Cleveland, W. S. (1984) "Graphical Methods for Data Presentation: Full Scale Breaks, Dot Charts, and Multibased Logging," *The American Statistician*, Vol. 38, No. 4, pp. 270-280.

Cleveland, W. S. (1985) *The Elements of Graphing Data*, Monterey, CA: Wadsworth.

Cleveland, W. S. (1993a) "A Model for Studying Display Methods of Statistical Graphics," *Journal of Computational and Graphical Statistics*, Vol. 2, No. 4, 323-343.

Cleveland, W. S. (1993b) *Visualizing Data*, Summit NJ: Hobart Press.

Courteau, J. B. (Editor) (1990), *Toxics in the Community, 1988 National and Local Perspectives*, EPA 560/4-90-017, Washington, DC.: U.S. Government Printing Office.
251.

Kosslyn, S. M. (1994) *Element of Graphic Design*, New York, NY: W. H. Freeman and Company.

Data analysis with graphical models: software tools

Wray L. Buntine, RIACS
NASA Ames Research Center
Mail Stop 269–2
Moffett Field, CA 94035–1000, USA
wray@kronos.arc.nasa.gov

Abstract

Probabilistic graphical models (directed and undirected Markov fields, and combined in chain graphs) are used widely in expert systems, image processing and other areas as a framework for representing and reasoning with probabilities. They come with corresponding algorithms for performing probabilistic inference. This paper discusses an extension to these models by Spiegelhalter and Gilks, plates, used to graphically model the notion of a sample. This offers a graphical specification language for representing data analysis problems. When combined with general methods for statistical inference, this also offers a unifying framework for prototyping and/or generating data analysis algorithms from graphical specifications. This paper outlines the framework and then presents some basic tools for the task: a graphical version of the Pitman-Koopman Theorem for the exponential family, problem decomposition, and the calculation of exact Bayes factors. Other tools already developed, such as automatic differentiation, Gibbs sampling, and use of the EM algorithm, make this a broad basis for the generation of data analysis software.

Introduction

This paper argues that the data analysis tasks of learning and knowledge discovery can be handled using graphical models [11]. This meta-level use of graphical models was first suggested by Spiegelhalter and Lauritzen in the context of learning probabilities for Bayesian networks. An extension of the standard graphical model is used here that allows this kind of learning to be represented. The extension is the notion of a *plate* introduced by Spiegelhalter and GilksGilks.etal.stat. Plates allow samples to be represented explicitly on the graphical model, and thus reasoned about. This makes data analysis problems explicit in much the same way that utility and decision nodes are used for decision analysis problems.

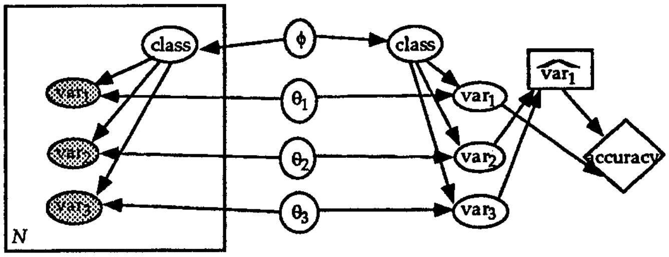


Figure 1: Simple unsupervised learning, with general prediction

Consider, for instance, Figure 1. This presents a situation where a mixture model with hidden variable *class* is used for subsequent prediction of var_1 from var_2 and var_3. The part to the left of the parameters θ and ϕ is the graphical representation of a sample, and the part to the right represents the prediction task. The value node, the diamond, indicates that subsequent prediction accuracy is the goal of learning, while the contents of the *plate* (the large box around the nodes for *class*, var_1, var_2 and var_3) indicates that a sample of N values of var_1, var_2 and var_3 are given, because they are shaded, while *class* is hidden, being unshaded. The plate indicates that its contents are replicated N times, yielding a product $\prod$ in the probability form. A legend for graphical models used in this paper appears in Figure 2.

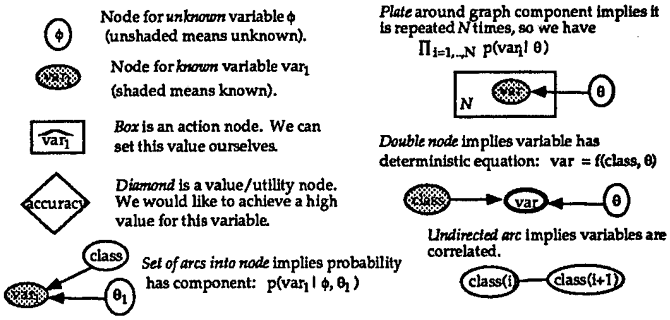

Node for *unknown* variable ϕ (unshaded means unknown).
Node for *known* variable var1 (shaded means known).
Box is an action node. We can set this value ourselves.
$\widehat{var_1}$
Diamond is a value/utility node. We would like to achieve a high value for this variable.
accuracy
class
Set of arcs into node implies probability has component: p(var1 | ϕ, θ1)
θ1
Plate around graph component implies it is repeated *N* times, so we have
Π i=1,..,N p(var1 | θ)
N
θ
Double node implies variable has deterministic equation: var = f(class, θ)
var
θ
Undirected arc implies variables are correlated.
class(i)
class(i+1)

Figure 2: A legend for graphical symbols

A general approach to the design of learning and data analysis algorithms now becoming widespread is one of engineering using principles of probability. An example is given in [3] where decision tree algorithms, made popular by the CART, ID3 and C4.5 programs, are developed from basic probability principles. The basic tools of probabilistic (Bayesian) inference used for this type of process are reviewed, for instance, by Tanner [10] and Kass and Raftery [9]: various exact methods, Markov chain Monte Carlo methods such as Gibbs sampling, the EM algorithm, and the Laplace approximation. With creative combination, these are able to address a wide range of data analysis problems. Gilks, Spiegelhalter and Thomas have taken this process a step further by developing a compiler that generates Gibbs samplers from graphical specifications [8]. This handles a surprisingly broad number of statistical tasks.

It is the thesis of this paper that these techniques are now sufficiently well developed so that software support can be provided for their use in data analysis problems. That is, we are now able to generate components of data analysis algorithms, and even entire algorithms themselves from high-level specifications. More details of this general capability can be found in [2]. A software generator needs two parts to make it work:

Language to specify problems: probabilistic graphical models (chain graphs [11]) extended with plates are used as a specification language. When augmented with specific functional forms such as the Gaussian and the logistic, this language is sufficient powerful to represent a broad range of problems across several fields: generalized linear models, feed-forward networks, Jordan and Jacobs mixture of experts, unsupervised learning of many different kinds, and hybrids of these models. A simple connectionist feed-forward network and its corresponding Bayesian network is given in Figure 3(a) and (b) respectively. The Bayesian network represents the feed-forward network using deterministic nodes and then tacks on an error model at the end of the network to indicate that the measured response variables are not deterministic functions of the inputs. The feed-forward network in this configuration therefore computes means of a Gaussian.

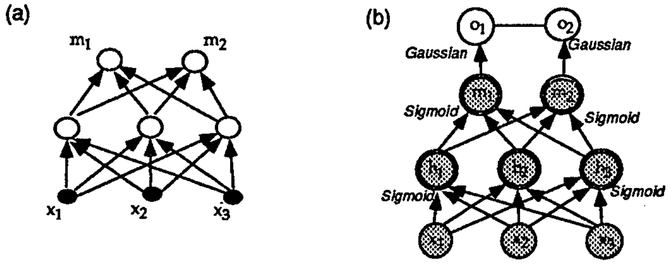


Figure 3: A simple feed-forward network: (a) in native form (b) as a DAG

Algorithm schemas: these are templates for high-level algorithms prior to code generation and compilation.

- Gilks *et al* [8] have developed general algorithms to perform Gibbs sampling on Bayesian networks with plates.
- Other algorithms such as conjugate gradient, Fisher's scoring method, or Laplace approximations [9] can be applied once first and second derivatives are calculated for model parameters.
- The automatic calculation of derivatives on structures is a well understood problem. In neural networks, this corresponds to the Back-propagation algorithm and its extensions for second derivatives. Likewise, the calculation of derivatives on probabilistic graphical models is an application of the chain rule for differentiation. Details appear in [2].
- The more general application of the EM algorithm for hidden variables is obvious.

Component libraries: Almond *et al.* [1] point out that parts of a graph, *components*, are often shared in a series of applications. Learning and data analysis are no different. One useful component is the generalized linear model which can include basis function sets for orthogonal polynomials or wavelets.

We can see that many parts of this ambitious plan, a software tool kit for data analysis, are already in place. The plan needs to be qualified, however. Proponents of Gibbs sampling, for instance, say that the design of an efficient sampler takes care and experience. Specific matrix forms might be used to advantage. It is often the case that some fine tuning is needed in algorithms. The aim here is to provide tools for software engineering, not complete packaged solutions.

One task that can never have direct software support is the design of an appropriate model with an appropriate prior. This is a knowledge elicitation problem. Techniques here are varied and range from careful choice of the representation to simplify elicitation, to techniques for working with components and libraries [1]. But the elicitation task still has to be done afresh with each different problem, except in those prototypical situations that are routinely addressed by standard statistical packages. While one might use a standard package in initial modeling, as the problem becomes better understood, specific requirements are needed that canned software may not provide. Of course, tools for software generation alleviate the modeling task greatly by providing rapid prototyping. Nevertheless, it is my view that a sizable burden in the Bayesian analysis of data is software engineering rather than the statistical analysis itself, and therefore software generators and support tools are both a realistic and important goal.

In this paper we discuss a few more pieces for this general software toolkit. The first is an algorithm for the decomposition of a chain graph with plates into independent components. This technique has been used to develop efficient algorithms for learning Bayesian networks from complete data [4]. The second contribution is some exact algorithms on graphical models with a single plate. Both these simplify calculation of the Bayes factor for a model, used widely in Bayesian methods [9].

The Bayes factor is the support given to model M_2 relative to model M_1 by the data *sample*.

$$Bayes\text{-}factor(M_2, M_1) = \frac{p(sample|M_2)}{p(sample|M_1)} .$$

We use the term *evidence* for the basic component,

$$evidence(M) = p(sample|M)$$

and consider its calculation throughout.

While these techniques can be used in many places in a learning toolkit, one interesting by-product is that they show how to develop algorithms for learning DAGs from complete data where the conditional distributions are in the exponential family, including mixtures of Gaussians, Poissons, discrete variables, etc. All that is required is a conjugate prior. While this capability should not be surprising —and perhaps the hardest part, appropriate priors, is left out—it is interesting that we can construct these algorithms automatically using the operations presented here. More recent work has focused on the development of priors and their use in the broader scheme of things.

Exact algorithms on graphs with plates

The removal of a plate from a graphical model requires conditions that are well known in statistics. The problem reduces to the existence of sufficient statistics giving a graphical version of the Pitman-Koopman Theorem from statistics.

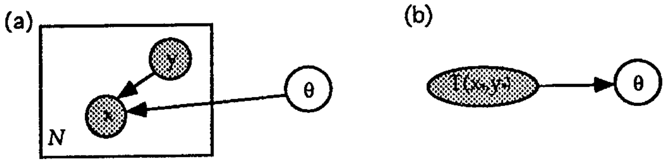


Figure 4: The generalized graph for plate removal

Comment 1 *(Plate removal). Consider the model M represented by the graphical model for a sample of size N given in Figure 4(a), where x is in the domain X and y is in the domain Y, both independent of θ, and both domains have components that are real valued or finite discrete. Let the conditional distribution for x given y, θ be $f(x|y, \theta)$, which is positive for all $x \in X$. If first derivatives exist w.r.t. all real valued components of x and y*[1]*, the plate removal operation applies for all samples $x_* = x_1, \ldots, x_N$, $y_* = y_1, \ldots, y_N$, and θ, as given in Figure 4(b), for some sufficient statistics $T(x_*, y_*)$ of dimension independent of N if and only if the conditional distribution for x given y, θ is in the exponential family, with form*

$$p(x|y,\theta,M) = \frac{h(x,y)}{Z(\theta)} \exp\left(\sum_{i=1}^{k} w_i(\theta) t_i(x,y)\right) , \quad (1)$$

for some functions w_i, t_i, h and Z and some integer k. In this case, $T(x_*, y_*)$ is an invertible function of the k averages

$$\frac{1}{N}\sum_{j=1}^{N} t_i(x_j) \quad : \quad i = 1, \ldots, k .$$

Graph decomposition

Learning problems can be decomposed into subproblems in some cases. For instance, consider the learning problem given in Figure 5 over two multinomial variables var_1 and var_2, and two Gaussian variables x_1 and x_2. For this problem we have specified two alternative models, model M_1 and model M_2. Model M_2 has an additional arc going from the discrete variable var_2 to the real valued variable x_1. We will use this subsequently to discuss local search of these models evaluated by their Bayes factor.

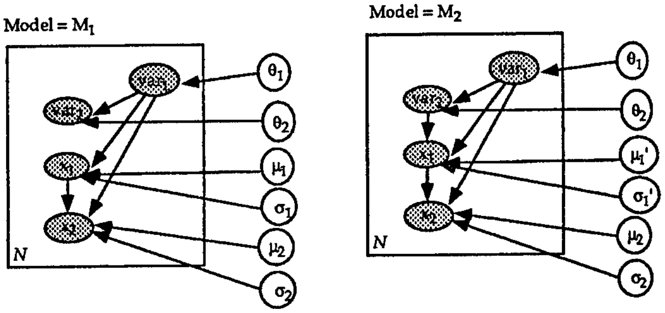


Figure 5: Two graphical models

A straight forward manipulation of the conditional distribution for this model yields, for model M_1, the conditional distribution given in Figure 6. When parameters, θ_1, θ_2, etc., are *a priori* independent, and their data

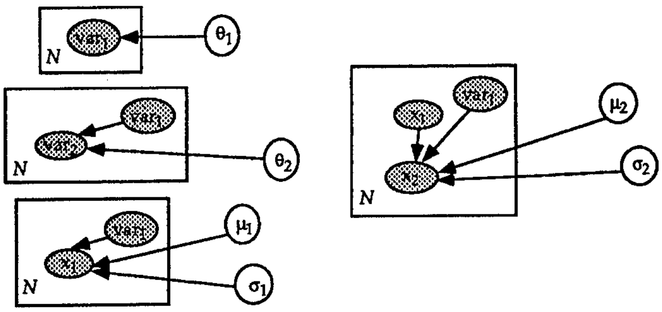


Figure 6: A simplification of model M_1

[1] I have yet to find a clear development of this. Usually, y isn't included in the classic treatment, but we need it here and it works.

likelihoods do not introduce cross terms between them, the parameters become *a posteriori* independent as well. This occurs for θ_1, θ_2, and the set $\{\mu_1, \sigma_1\}$. This model simplification also implies the evidence for model M_1 decomposes similarly. Denote the sample of the variable x_1 as $x_{1,*} = x_{1,1}, \ldots, x_{1,N}$, and likewise for var_1 and var_2, etc. In this case, we get,

$$\begin{aligned} evidence(M_1) &= p(var_{1,*}|M_1)\, p(var_{2,*}|var_{1,*}, M_1) \quad (2) \\ &\quad p(x_{1,*}|var_{1,*}, M_1)\, p(x_{2,*}|x_{1,*}, var_{1,*}, M_1)\ . \end{aligned}$$

The evidence for model M_2 is similar except that the posterior distribution of μ_1 and σ_1 is replaced by the posterior distribution for μ'_1 and σ'_1.

This result is general, and applies to both DAGs, undirected graphs, and more generally to chain graphs. Similar results results are covered by Dawid and Lauritzen [7] for a family of models they call hyper-Markov. The general result described above is an application of the rules of independence applied to plates. This uses a notion of local dependence, which is called the Markov blanket. The Markov blanket is a node's parents, children, and the children's parents. If deterministic nodes are involved, the definition requires a bit more care [2].

To perform the simplification depicted in Figure 6, it is sufficient then to find the finest partitioning of the model parameters such that they are independent. The decomposition in Figure 6 represents the finest such partition of model M_1. The evidence for the model will then factor according to the partition, as given for model M_1 in Equation (2). For this task we have the following theorem.

Theorem 1 *(Decomposition). A model M is represented by a chain graph G with plates and no deterministic nodes. Let the variables in the graph be X. We have P possibly empty subsets of the variables X, X_i for $i = 1, \ldots, P$ such that $unknown(X_i)$ is a partition of $unknown(X)$. This induces a decomposition of the graph G into P subgraphs G_i where:*

- *the graph G_i contains the nodes X_i and any arcs and plates occurring on these nodes; and*
- *the potential functions for cliques in G_i are equivalent to those in G.*

The induced decomposition represents the unique finest equivalent independence model to the original graph if and only if X_i for $i = 1, \ldots, P$ is the finest collection of sets such that, when ignoring plates, for every unknown node u in X_i, its Markov blanket is also in X_i. This finest decomposition takes $O(|X|^2)$ to compute. Furthermore, the evidence for M now becomes a product over each subgraph,

$$evidence(M) = f_0 \prod_i f_i(known(X_{i,*}))\ , \qquad (3)$$

for some functions f_i (given in the proof).

In some cases, the functions f_i have a clean interpretation: they are equal to the evidence for the subgraphs. This result can be obtained from the following corollary.

Corollary 1.1 *In the context of Theorem 1 where there are no deterministic nodes, suppose there exists a set of chain components τ_j from the graph ignoring plates such that $X_j = \tau_j \cup parents(\tau_j)$, where $unknown(parents(\tau_j)) = \emptyset$. Then*

$$f_j(known(X_{j,*})) = p(known(\tau_j)_*|parents(\tau_j)_*, M)\ .$$

When deterministic nodes exist, this is altered by redefining the notion of parent [2].

If we denote the j-th subgraph by model M_j, then the probability term in the corollary is the conditional evidence for model M_j given $parents(\tau_j)_*$. Denote by M_0 the subgraph on known variables induced by $cliques_0$ (as given in the proof [2]). If the condition of Corollary 1.1 holds for M_j for $j = 0, 1, \ldots, P$, then it follows that the evidence for the model M is equal to the product of the evidence for each subgraph.

$$evidence(M) = \prod_{i=0}^{P} evidence(M_i)\ . \qquad (4)$$

This holds in general if the original graph G is a DAG, as used in learning DAGs [4].

Corollary 1.2 *Equation (4) holds if the parent graph G is a DAG with plates.*

In general, we might consider searching through a family of graphical models. To do this we can use standard methods such as local search or numerical optimization to find high posterior models, or Markov chain Monte Carlo methods to select a sample of representative models [2]. To do this, we first show how to represent a family of models. Figure 7, for instance, is similar to models of Figure 5 except that some arcs are hatched. We use this to indicate that these arcs are optional. To instantiate a hatched arc they can either be removed, or replaced with a full arc. This graphical model then represents many different models, for all 2^4 possible instantiations of the arcs. Prior probabilities for these models could

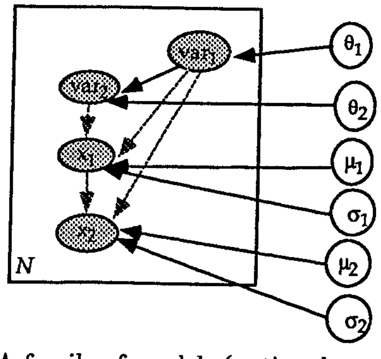


Figure 7: A family of models (optional arcs hatched)

be generated using a scheme such as in [4, p54] where a prior probability is assigned by a domain expert for the inclusion of each arc, and the prior for a full model found by multiplication. The family of models given by Figure 7 includes those of Figure 5 as instances. During search or sampling, an important property is the Bayes factor for the two models, $Bayes\text{-}factor(M_2, M_1)$. Because of the decompositions above, the Bayes factor can be found by only examining component Bayes factors for nodes whose parents have changed between models M_1 and M_2. The difference here is the model for the variable x_1.

$$Bayes\text{-}factor(M_2, M_1) = \frac{p(x_{1,*}|var_{1,*}, var_{2,*}, M_2)}{p(x_{1,*}|var_{1,*}, M_1)}$$

That is, the Bayes factor can be computed from only considering the models involving μ_1, σ_1 and μ_1', σ_1'.

This incremental modification of evidence, Bayes factors, and finest decompositions is also general, and follows directly from the independence test. It has been used in fast learning algorithms for DAGs [4]. This is developed below for the case of directed arcs and non-deterministic variables.

Lemma 1 *For a graph G in the context of Theorem 1 with no deterministic nodes, we have two variables U and V such that U is given. Consider adding/removing a directed arc from U to V. We update the finest decomposition of G as follows: There is a unique subgraph containing the unknown variables in parents(chain-component(V)). To this subgraph add/delete an arc from U to V, and add/delete U to the subgraph if required.*

We can therefore add shaded non-deterministic parents at will to nodes in a graph and the finest decomposition remains unchanged except for a few additional arcs. The use of hatched arcs in these contexts therefore causes no additional trouble to the decomposition process. That is, we form the finest decomposition for a graph with plates and hatched directed arcs as if the arcs were normal directed arcs, and the evidence is adjusted during the search by adding the different parents as required.

Bayes factors for the exponential family

The above results are useful, but to make use of them automatically we need to be able to generate Bayes factors or evidence for models. It generally holds that if a likelihood is in the exponential family, then the posterior distribution for the model parameters is also in the exponential family, although it is only really useful when the normalizing constant is readily computed. This holds for the Dirichlet, the conjugate to a multinomial, and the Gaussian-Wishart, the conjugate to a Gaussian. We give the results here.

Let the normalizing constant for the conjugate distribution for Comment 1 be $Z_\theta(\tau)$, and let the normalizing constant for the distribution be $Z_1(\theta)Z_2$ where Z_2 is a constant part independent of θ, then the Bayes factor can be readily computed. This is a common trick used widely by Bayesians, however, I have never seen it stated explicitly.

Lemma 2 *Consider the context of Comment 1. Then the model likelihood or evidence, given by $evidence(M) = p(x_1, \ldots, x_N|y_1, \ldots, y_N, M)$, can be computed as:*

$$\begin{aligned} evidence(M) &= \frac{p(\theta|\tau)\prod_{j=1}^N p(x_j|y_j, \theta)}{p(\theta|\tau')} \\ &= \frac{Z_\theta(\tau')}{Z_\theta(\tau)Z_2^N} . \end{aligned}$$

References

[1] R.G. Almond, J.M. Bradshaw, and D. Madigan. Reuse and sharing of graphical belief network components. In Cheeseman and Oldford [5], pages 113–122.

[2] W. Buntine. Operations for learning using graphical models. *Journal of Artificial Intelligence Research*, 1994. to appear.

[3] W.L. Buntine. Learning classification trees. In D.J. Hand, editor, *Artificial Intelligence Frontiers in Statistics*, pages 182–201. Chapman & Hall, London, 1991.

[4] W.L. Buntine. Theory refinement of Bayesian networks. In B.D. D'Ambrosio, P. Smets, and P.P. Bonissone, editors, *Uncertainty in Artificial Intelligence: Proceedings of the Seventh Conference*, Los Angeles, CA, 1991.

[5] P. Cheeseman and R.W. Oldford, editors. *Selecting Models from Data: Artificial Intelligence and Statistics IV.* Springer-Verlag, 1994.

[6] G.F. Cooper and E.H. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–348, 1992.

[7] A.P. Dawid and S.L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21(3):1272–1317, 1993.

[8] W.R. Gilks, A. Thomas, and D.J. Spiegelhalter. A language and program for complex Bayesian modelling. *The Statistician*, 1993.

[9] R.E. Kass and A.E. Raftery. Bayes factors and model uncertainty. Technical Report #571, Department of Statistics, Carnegie Mellon University, PA, 1993. Submitted to Jnl. of American Statistical Association.

[10] M.A. Tanner. *Tools for Statistical Inference.* Springer-Verlag, New York, second edition, 1993.

[11] J. Whittaker. *Graphical Models in Applied Multivariate Statistics.* Wiley, 1990.

Relative power of Smirnov and Wilcoxon exact tests in two-sample ordered categorical data

Joan F. Hilton
Department of Epidemiology and Biostatistics, University of California,
San Francisco, CA 94143-0560, U.S.A.

Cyrus R. Mehta
Cytel Software Corporation, Cambridge, MA 02139, and Department of
Biostatistics, Harvard School of Public Health, Boston, MA 02115, U.S.A.

Nitin R. Patel
Cytel Software Corporation, Cambridge, MA 02139, and
Indian Institute of Management, Ahmedabad 380 056, India

Abstract

In two-sample studies with ordinal responses, the Wilcoxon rank-sum test is commonly chosen to test equality of the distributions, $H_0: F_1 = F_2$, in spite of its being a test of the specific hypothesis of location-shift between the distributions. Unless a specific alternative is hypothesized, use of an omnibus test instead should maximize power. We compare the power of the exact tests based on the omnibus classical Smirnov statistic with that based on the Wilcoxon rank-sum statistic under various alternatives, including shift in location. To compute exact power we use the methods described by Hilton and Mehta (1993) and Mehta, Patel and Tsiatis (1984). These algorithms are especially useful in evaluating the Smirnov test because its asymptotic non-null distribution has not been defined. Specific examples as well as results of a simulation study are presented.

Introduction

When two-sample data with categorical responses are analyzed, if the responses are ordinal then the Wilcoxon rank-sum test (Wilcoxon, 1945) is commonly chosen to test equality of the distributions, $H_0: F_1 = F_2$. The Wilcoxon rank-sum statistic is specifically sensitive to the hypothesis

$$H_0: F(x) = F(x - \Delta),$$

where Δ represents a location-shift between the distributions of responses. However, especially in categorical data, little may be known about the types of differences that occurs between distributions, in which case an omnibus test should generally increase power. For example, the responses may differ in scale, τ, as well as in location,

$$H_0: F(x) = F\left(\frac{x - \Delta}{\exp(\tau)}\right)$$

A candidate omnibus test for two-sample ordered categorical data is that based on the Smirnov statistic (Smirnov, 1939).

Attempting to obtain high power, Eplett (1982) proposed a statistic that is the sum of the Wilcoxon and Smirnov statistics and evaluated its power under location and scale alternatives. He showed that "for light-tailed distributions" his test becomes progressively more powerful compared with the Smirnov test as the scale-change part of the hypothesis becomes more dominant. When one of the two distributions was uniform over $[0, 1]$, the power of the tests based on these two statistics were similar for $m = n = 50$. More recently, O'Brien (1988) and Blair and Morel (1992) evaluated four tests in the presence of location and scale changes in continuous data: Wilcoxon's test, Student's t test, and O'Brien's generalized versions of these tests (1988). The generalizations were defined to increase the sensitivity of the tests to scale changes. However, Blair and Morel (1992) found that "heterogeneity of patient response (scale change) does not always lead to power advantage for the unconditional generalized tests." Thus, at least in the continuous response data realm, the need for a test that is sensitive to a broad range of alternatives has not yet been satisfied.

Here, we compare the power of the Smirnov test with that of the Wilcoxon test, the standard in practice, under alternatives that include changes in location and/or scale. The underlying data are ordered categorical. Because asymptotic power formulæ that account for categorical data do not exist, we evaluate the exact power of these tests. Mehta, Patel and Tsiatis (1984) and Hilton and Mehta (1993) described methods for testing or finding power of exact tests which are illustrated via the Wilcoxon statistic. Hilton, Mehta and Patel (1994) and Nikiforov (1994) have recently reported algorithms for conducting exact Smirnov tests for continuous or categorical data. Method are presented for computing exact power and for modeling alternatives of interest between the distributions of the two groups. Finally, we explore the relative power of the Wilcoxon and Smirnov tests against a range of these alternatives.

Methods

Let $\mathbf{x} = (x_1, \ldots, x_K)$, $\sum x_i = m$, and $\mathbf{x}' = (x'_1, \ldots, x'_K)$, $\sum x'_i = n$, represent two samples of responses from multinomial distributions with parameters $(\pi_1, \ldots, \pi_K)$, $\sum \pi_j = 1$, and $(\pi'_1, \ldots, \pi'_K)$, $\sum \pi'_j = 1$, respectively. Denote the combined data by $t_j = x_j + x'_j$, $j = 1, \ldots, K$, where K is the number of distinct categories in the combined sample. Then the probability of a particular permutation of the data, conditional on $\mathbf{t} = (t_1, \ldots, t_K)$ is given by the generalized hypergeometric distribution (Lehmann, 1975),

$$Pr\{\mathbf{X} = \mathbf{x}|\mathbf{t}\} = \frac{\left(m! \prod_{j=1}^{K} \frac{\pi_j^{x_j}}{x_j!}\right)\left(n! \prod_{j=1}^{K} \frac{{\pi'_j}^{t_j - x_j}}{(t_j - x_j)!}\right)}{m!n! \sum_{\mathbf{y}} \prod_{j=1}^{K} \frac{\pi_j^{y_j} {\pi'_j}^{t_j - y_j}}{y_j!\,(t_j - y_j)!}},$$

where $\mathbf{x}, \mathbf{y} \in \Gamma_{\mathbf{t}}$, the set of all such permutations:

$$\Gamma_{\mathbf{t}} = \{\mathbf{x} : \sum_{j=1}^{K} x_j = m, \sum_{j=1}^{K} x'_j = n, \text{ and } \mathbf{x} + \mathbf{x}' = \mathbf{t}\}.$$

Under an alternative hypothesis H_A, π' specifies a particular alternative of interest. Then the power of, say, the test based on the Wilcoxon statistic, is

$$\begin{aligned} \beta_{\mathbf{t}}(w) &= Pr\{W \geq w|\mathbf{t}; H_A\} \qquad (1) \\ &= \sum_{\mathbf{x} \in \Gamma_{\mathbf{t}}(w)} Pr\{\mathbf{X} = \mathbf{x}|\mathbf{t}; H_A\}, \end{aligned}$$

where $\Gamma_{\mathbf{t}}(w) = \{\mathbf{x} \in \Gamma_{\mathbf{t}} : \sum_j a_j x_j \geq w,\ j = 1, \ldots, K\}$. Similarly, the test could be based on the Smirnov statistic, in which case the critical region would be $\Gamma_{\mathbf{t}}(s) = \{\mathbf{x} \in \Gamma_{\mathbf{t}} : \max_j[\hat{F}_m(j) - \hat{F}_n(j)] \geq s,\ j = 1, \ldots, K\}$, where $\hat{F}_m(j) = \frac{1}{m}\sum_{i=1}^{j} x_i$ – the empirical distribution function.

Since power can be computed conditionally for all margins $\mathbf{t}$ (1), exact unconditional power can be obtained as the expected value of these terms,

$$\beta(w) = \sum_{\mathbf{t} \in \Omega} \beta_{\mathbf{t}}(w) Pr\{\mathbf{T} = \mathbf{t}; H_A\}, \qquad (2)$$

where $\Omega = \{\mathbf{t} : \sum t_j = m + n\}$ and $Pr\{\mathbf{T} = \mathbf{t}; H_A\} = \sum_{\mathbf{x} \in \Gamma_{\mathbf{t}}} Pr\{\mathbf{X} = \mathbf{x}, \mathbf{X}' = \mathbf{x}'; H_A\}$. In theory obtaining (2) is clearly not difficult, but in practice it is because the size of Ω can be quite large. For example, for $K = 5$ and $m + n = 50$, Ω contains 316,251 distinct vectors $\mathbf{t}$.

To reduce the computational burden one can instead estimate exact power from a sample of Ω, given m and n. Hilton and Mehta (1993) described a Monte Carlo estimator of exact power,

$$\hat{\beta}(w) = \frac{1}{N} \sum_{i=1}^{N} \beta_{\mathbf{t}_i}(w). \qquad (3)$$

and reported its high efficiency relative to the usual Monte Carlo estimator when using the Wilcoxon statistic in 5-category data.

Modeling alternatives

To account for the ordering of the responses, for group 1 define the cumulative probability of responding in categories 1 through j as $\gamma_j = \pi_1 + \pi_2 + \cdots + \pi_j$, and define the corresponding cumulative number of subjects responding in categories 1 through j as $m_j = m_{j-1} + x_j$, $j = 1, \ldots, K$, where $m_0 \equiv 0$ and $m_K \equiv m$. For group 2 define γ'_j and n_j, $j = 0, \ldots, K$, analogously. The cumulative probabilities are useful in specifying the distributions under alternative hypotheses.

To simplify the problem of specifying alternatives in ordered categorical data, we find an extension of the proportional odds model (McCullagh, 1980) useful:

$$\text{logit}(\gamma'_j) = \frac{\text{logit}(\gamma_j) - \Delta}{\exp(\tau)},\ j = 1, \ldots, K - 1, \qquad (4)$$

where $\Delta, \tau \in (-\infty, \infty)$, and $\Delta = 0$, $\tau = 0$ represents the null case. The model reduces the $2(K-1)$ possible parameters to $K - 1$ nuisance parameters, γ_j, $j = 1, \ldots, K - 1$, and two parameters of interest, Δ and τ.

The nuisance parameters might represent, for example, the distribution of the control group whose values can be obtained from previous research.

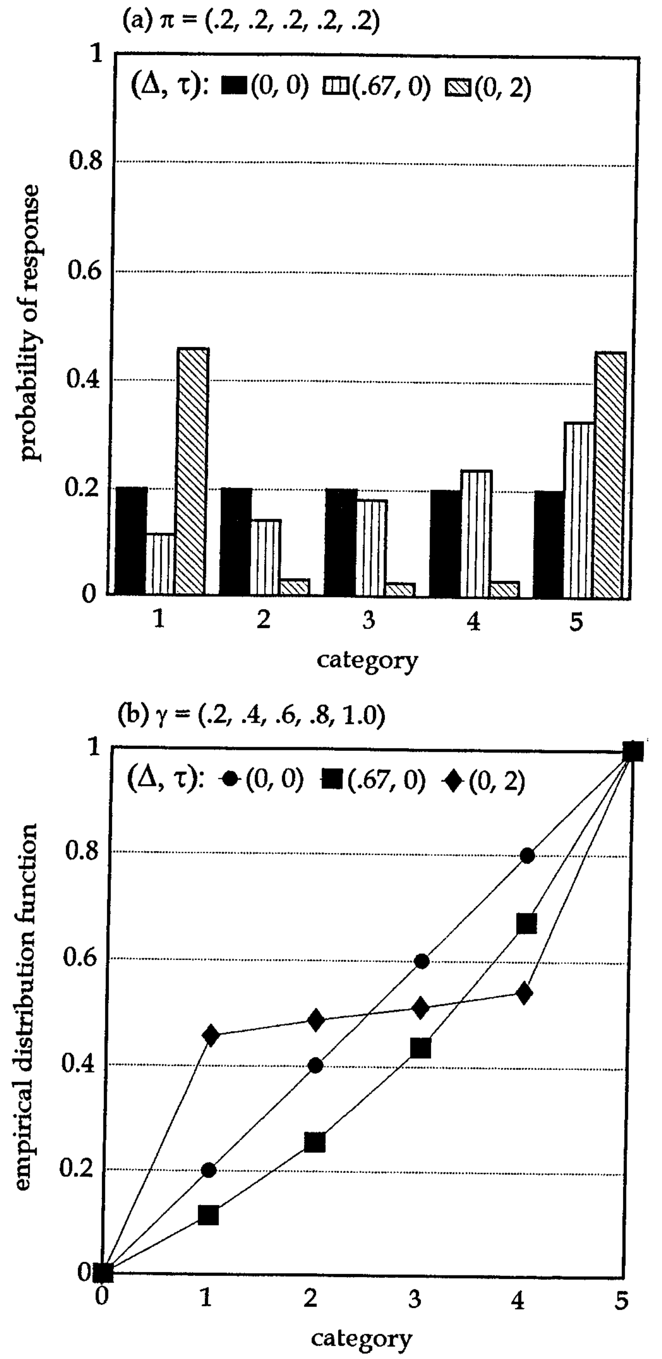


Figure 1. Distributions arising from equation (4) for three combinations of (Δ, τ) as a function of (a) $\pi = (.2, .2, .2, .2, .2)$ and (b) $\gamma = (.2, .4, .6, .8, 1.0)$.

Figure 1 illustrates some distributions that can arise from this model. Clearly, a rich field of alternatives can be specified through such a model, against some of which the Wilcoxon test may be more sensitive (Δ changes) and others the Smirnov test may be more sensitive (τ changes).

Example

Lesaffre, Scheys, Frölich and Bluhmki (1993) described the problem of calculating sample size in studies with bounded outcome scores. Their responses fell into 21 categories, obtained by collapsing a continuous 0 – 100 scale, with high probabilities in the first and last categories. They note that when the data have a U-shaped or J-shaped distribution, the assumptions underlying Lehmann's method of determining power via the Wilcoxon statistic are not met. This method indicates that 120 subjects per group are needed to detect a standardized difference of .38 with 80% power using a two-sided .05-level Wilcoxon statistic; we add that 95 subjects per group are needed using a one-sided test.

Their 21-point scale data are shown in Figure 2. Using the estimator described in (3) with $N = 5$, we estimated that the exact two-sided Wilcoxon power was $47.1 \pm 1.1\%$ and Smirnov power was $44.1 \pm 1.1\%$) — far less than the 80% obtained by the asymptotic approximation.

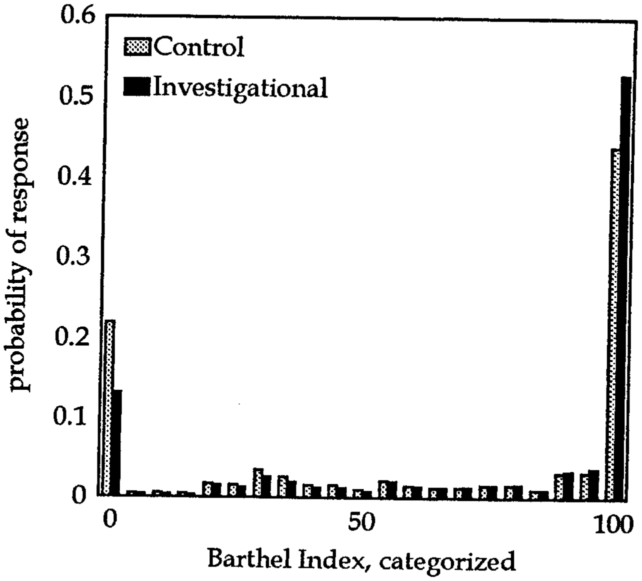


Figure 2. Distributions of Barthel's Index scores of control and treated subjects. (Modified from Lesaffre, Scheys, Frölich and Bluhmki (1993).)

Relative power

We compared more generally the unconditional power of the .05-level one-sided exact Wilcoxon and Smirnov tests against location (Δ) and/or scale (τ) alternatives in ordered categorical data. Group 1 data were generated by transforming $m = 50$ (or 25) uniform[0,1] random variates into 5-category multinomial(m, π) counts, where $\pi = (.2, .2, .2, .2, .2)$ ($\gamma = (.2, .4, .6, .8, 1.0)$). Group 2 data with $n = 50$ (or 75) were generated similarly using (4) to specify γ'. We evaluated Δ=(0,.1(.2)1.3) by τ= (0,.2,.4). These alternatives lead to large values of the Wilcoxon and Smirnov statistics when testing $H_A : \gamma_j < \gamma'_j$, $j = 1, \ldots, K-1$. For each combination of parameters, $N = 100$ two-sample data sets were simulated.

Figure 3 displays the relative power of the Smirnov to the Wilcoxon test in (a) balanced samples ($m = 50, n = 50$) and (b) unbalanced samples ($m = 25, n = 75$), as a function of Δ, for three values of τ. The relative power curves when $\tau = 0$ indicate that against location alternatives the Smirnov test was less powerful than the Wilcoxon test in both balanced and unbalanced samples. The relative power of the Smirnov test was lowest at $\Delta = 0$, which demonstrated that it was more conservative (Smirnov size = 3.4 ± .07%; Wilcoxon size = 5.0 ± .00%). As Δ moved away from zero, the relative power of the Smirnov test increased. At $\Delta = .9$, Wilcoxon power was 70% or 81%, depending on balance of samples, and Smirnov power achieved .84 – .86 times as much power.

As τ moved away from the null case, both tests had greater power than in the $\tau = 0$ case at all values of Δ. At $\Delta = .9$, $\tau = .2$, Wilcoxon power was 77% or 88%, and Smirnov power achieved .88 – .91 times these levels.

For $\tau = .4$, in the absence of location changes ($\Delta = 0$) the Smirnov test had substantially greater power. The relative power decreased with the increasing influence of the location parameter, but remained $\geq .95$ for all Δ. At $\Delta = .9$, $\tau = .4$, Wilcoxon power was 84% – 92%, and Smirnov power was .98–1.02 times as high. As expected, both tests generally had greater power in balanced than imbalanced samples. In addition, balance in sample sizes favored the Smirnov test.

Conclusions

The Wilcoxon rank-sum statistic is commonly used to analyze ordered categorical data. However, we believe that it should not be used without careful consideration of the alternative hypothesis of interest, since it was specifically designed to test a location-shift between distributions. We compared the power of exact tests based on the Wilcoxon rank-sum statistic and the Smirnov statistic, focusing on the setting that is optimal for the Wilcoxon since it is the "standard" for 2 × K ordered categorical data. We estimated the relative power of the

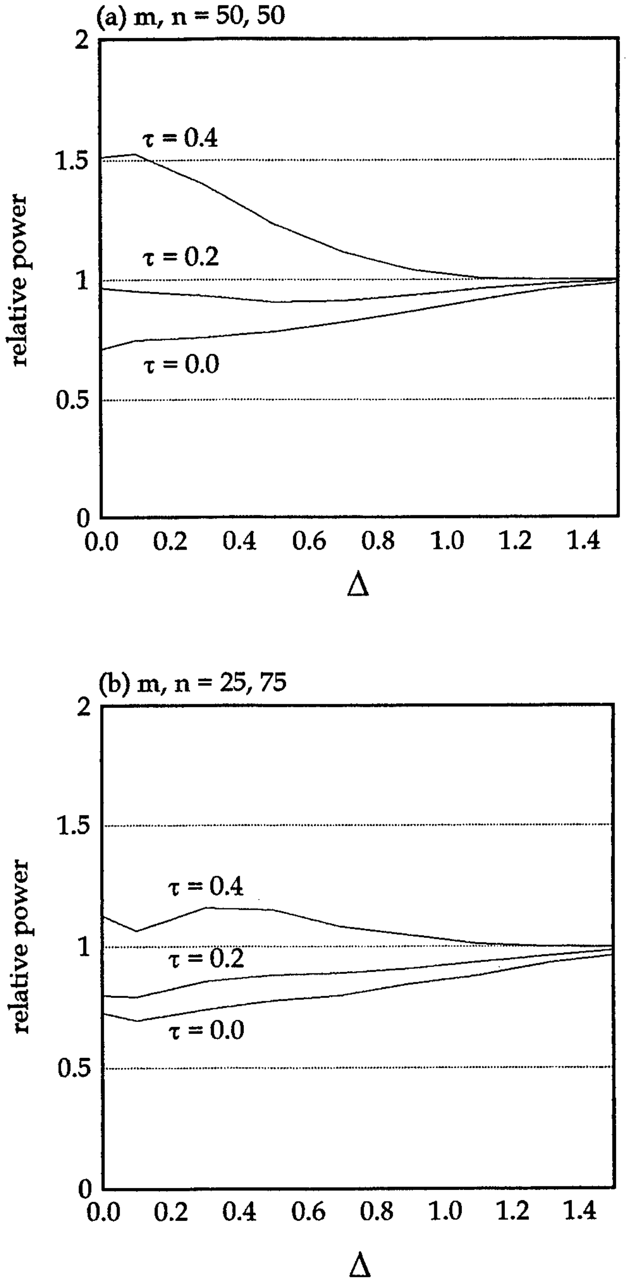


Figure 3. Relative power of Smirnov to Wilcoxon test, as a function of location (Δ) and scale (τ) changes between distributions, in (a) balanced and (b) unbalanced samples. Nuisance parameters are $\pi = (.2, .2, .2, .2, .2)$.

tests against location-shifts, with and without inclusion of local scale alternatives, using the method of Hilton and Mehta (1993).

Under the null hypothesis of no difference between two distributions, the Smirnov test was more conservative. This is because its support, or reference set, is more discrete than that of the Wilcoxon statistic.

Against location-shifts alone, the setting in which the Wilcoxon test is optimal, the Wilcoxon test was substantially more powerful. In contrast, the Smirnov test was very sensitive to scale changes alone, while the Wilcoxon test had negligible power in this setting.

For location-shifts in the presence of small scale changes, there was little difference in the power of these two tests. As the influence of the scale effect increased, so did the relative power of the Smirnov test. Without information on how two ordered categorical distributions differ, the test based on the Smirnov statistic is the safer choice. Our results held for two very different definitions of the nuisance parameters (only one shown here) and in balanced and unbalanced samples; they were strong enough to suggest that they hold fairly generally.

The choice of the vector of nuisance parameters, representing a control group, can often be guided by previous research. Specification of the vector of parameters in an experimental group can be more difficult; we proposed using a model, such as the proportional odds model, so that only location and/or scale differences between the distributions need be specified. The choice of which model to use is somewhat arbitrary. Its selection is akin to choosing to base a power calculation for 2 × 2 data on either the difference between binomial probabilities or on the odds ratio. Like the influence of the nuisance parameters, the impact of the model on the alternatives is much less than the impact of the location and/or scale parameters.

In an example data set with an 11-point ordinal response and large probabilities in the extreme categories, we showed that Lehmann's asymptotic approximation can provide a very poor estimate of the power of the Wilcoxon test for ordered categorical data. We don't attempt to generalize this finding, but rather use it to illustrate that Lehmann's approximation should be applied to categorical data with caution. Another drawback of asymptotic methods for power calculations is that the Wilcoxon statistic can be used but the Smirnov cannot – because its asymptotic non-null distribution has not been defined. In contrast, our exact method accommodates a variety of statistics, including the Wilcoxon rank-sum and omnibus Smirnov statistics, and is most efficient when samples are small and response distributions are discrete.

In conclusion, if two ordered categorical distributions differ at all in scale, with or without differences in location, then the Smirnov statistic should be used for designing and analyzing a study of the hypothesis of equality of distributions.

References

Blair, R.C. and Morel, J.G. (1992). On the use of the generalized *t* and generalized rank-sum statistics in medical research. *Statistics in Medicine* **11**:491 – 501.

Eplett, W.J.R. (1982). The distributions of Smirnov type two-sample rank tests for discontinuous distribution functions. *J. R. Statist. Soc.* B **44**:361–369.

Hilton, J.F., Mehta, C.R., and Patel, N.R. (1994). An algorithm for conducting exact Smirnov tests. *Computational Statistics and Data Analysis*, **17**:351 – 361.

Hilton, J.F. and Mehta, C.R. (1993). Power and sample size calculations for exact conditional tests with ordered categorical data. *Biometrics* **49**:609 – 616.

Lehmann, E.L. (1975). *Nonparametrics: Statistical Methods Based on Ranks.* San Francisco: Holden-Day, Inc.,

Lesaffre, E., Ilse, S., Frölich, J., and Bluhmki, E. (1993) Calculation of power and sample size with bounded outcome scores. *Statistics in Medicine* **12**:1063 – 1078.

Mehta, C.R., Patel, N.R., and Tsiatis, A.A. (1984). Exact significance testing to establish treatment equivalence with ordered categorical data. *Biometrics* **40**:819 – 825.

Nikiforov, A.M. (1994). Exact Smirnov two-sample tests for arbitrary distributions. *Applied Statistics* **43**:265 – 269.

O'Brien, P.C. (1988). Comparing two samples: extensions of the *t*, rank-sum and log-rank tests. *Journal of the American Statistical Association* **83**:52 – 61.

Smirnov, N.V. (1939). On the estimation of the discrepancy between empirical distribution curves for two independent samples. *Bulletin de l'Université de Moscou, Série internationale (Mathématiques)* **2**:3 – 14.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics* **1**:80 – 83.

A SIMULATION STUDY OF SOME RANK TESTS FOR INTERACTION IN TWO-WAY LAYOUTS

Guang-hwa Chang, Youngstown State University
Department of Mathematics, Youngstown State University, Youngstown, OH 44555

In this investigation, two versions of the F-statistic analogue of aligned rank test for interaction in two-way layouts are studied along with the classical F-test and the rank transform test. The Wilcoxon and the normal score functions are both considered in the study. The results from extensive simulation studies indicate that the aligned rank tests have better performance in general. None of these tests performs well for the Cauchy distribution. The use of the normal rank score function reduces the inflation in the Type I error rate of the rank transform statistic.

1 INTRODUCTION

In the traditional approach for testing main effects in two-way layouts, the existence of interaction needs to be tested first. In addition to the classical F-test, several nonparametric alternatives have been proposed by Mehra and Sen (1969), Mehra and Smith (1970), Bhapkar and Goré (1974), and Mansouri and Govindarajulu (1990). The aligned rank methods studied by McKean and Hettmansperger (1976), Adichie (1978), and Chiang and Puri (1984) can also be used to form tests for interaction.

The rank transform (RT) method consists of replacing the observations with their rank among the entire data set and performing the standard parametric analysis of variance (ANOVA) test to these ranks. Conover and Iman (1981) has suggested that the RT approach can be applied in a variety of circumstances such as analysis of experiment designs, multiple regression, cluster analysis, discriminant analysis. This type of testing procedure has many advantages over other procedures for its less strict distributional assumptions, greater power and also it is simple to apply because of the existing computer software for parametric tests. In the simulation studies by Iman, Hora and Conover (1984), they show that this procedure has excellent power properties for testing main effects in two-way layouts without interaction. Hora and Conover (1984) has also found the limiting null distribution of the usual F statistic when applied to ranks for testing main effects in two-way layouts without interaction. The simulation results of Blair, Sawilowsky and Higgins (1987) show that the RT statistic is inappropriate for testing interaction in two-way layouts. Under the assumption of normality and when both main effects are present, they found a severe inflation in Type I error rates as main effects are large.

The reason for the RT technique to be so attractive is their simplicity, since the classical F-test is available in almost every statistical package. In this paper, two statistics, aligned RT statistic and modified aligned RT statistic, that have the same features of the RT statistic are studied. They are more powerful and robust then the RT statistic and are referred to as the aligned rank transform statistics. Through the use of the alignment technique, the inflation in Type I error rates is overcome. The formulas for these aligned RT statistics are presented in section 2. In section 3, tables are generated for Type I error rates and power analysis. It is concluded that the aligned rank transform test has the most robust test among the group of tests. None of these tests performs well for the Cauchy distribution.

2 THE TEST STATISTICS

Let X_{ijk} denotes the kth random observation from the (i,j)th cell follow the fixed effects model:

$$X_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \epsilon_{ijk} \qquad (2.1)$$

$$i = 1, \cdots, r; \; j = 1, \cdots, c; \; k = 1, \cdots, n;$$

where $r \cdot c \cdot n = N, \mu$ is the overall mean, α_i and β_j are the ith row and the jth column main effects, respectively, γ_{ij} is interactions between the ith row and jth column, and ϵ_{ijk} are independent and identically distributed random variables having a continuous distribution function $F(\cdot)$. We wish to test the null hypothesis

$$H_0 : \gamma_{ij} = 0, \qquad \forall\, i, j,$$

against the alternative hypothesis

$$H_a : \gamma_{ij} \neq 0, \qquad \text{for at least one } (i,j).$$

The classical F-statistic is given by

$$F = \frac{MSINT}{MSE}, \qquad (2.2)$$

where

$$MSINT = n\sum_{i=1}^{r}\sum_{j=1}^{c}(\bar{X}_{ij\cdot}-\bar{X}_{i\cdot\cdot}-\bar{X}_{\cdot j\cdot}+\bar{X}_{\cdots})/(r-1)(c-1), \qquad (2.3)$$

$$MSE = \sum_{i=1}^{r}\sum_{j=1}^{c}\sum_{k=1}^{n}(X_{ijk}-\bar{X}_{ij\cdot})^2/rc(n-1), \qquad (2.4)$$

the combination of bar and dot notation means that the values are averaged over the subscript(s).

Let R_{ijk} denote the rank of X_{ijk} among $X_{111},\cdots, X_{11n},\cdots, X_{rcn}$. The RT statistic F is obtained from (2.2) by replacing X_{ijk} with the ranks or the rank scores $a_N(R_{ijk})$. We consider the rank score functions that satisfy the following general assumptions:

- The scores $a_N(i)$ are generated by a non-constant and square integrable function ϕ defined on $(0,1)$, in one of the following ways:

  $$a_N(i) = \phi[\frac{i}{N+1}], \text{ or, } a_N(i) = E[\phi(U_{i,N})],$$

  where $1 \leq i \leq N$, and $U_{i,N}$ is the ith order statistic in a sample of size N from a uniform distribution defined on $(0,1)$.

- The score generating function $\phi(u)$ on $(0,1)$ is such that

  $$0 < \sigma^2(\phi) = \int_0^1 [\phi(u)-\bar{\phi})^2 du < \infty, \bar{\phi} = \int_0^1 \phi(u)du.$$

Let $\{\hat{\alpha}_i\}_{i=1}^{r}$ and $\{\hat{\beta}_j\}_{j=1}^{c}$ denote some consistent estimators of $\{\alpha_i\}_{i=1}^{r}$ and $\{\beta_j\}_{j=1}^{c}$ under H_0, respectively, such that $N^{1/2}(\hat{\alpha}_i - \alpha_i)$ and $N^{1/2}(\hat{\beta}_i - \beta_i)$ are bounded in probability for every i and j. Let the aligned rank $\hat{R}_{ijk}$ be the rank of $X_{ijk} - \hat{\alpha}_i - \hat{\beta}_j$ among $X_{111} - \hat{\alpha}_1 - \hat{\beta}_1, \cdots, X_{rcn} - \hat{\alpha}_r - \hat{\beta}_c$. The the aligned RT statistic F_a has the form (2.2) by replacing X_{ijk} with $a_N(\hat{R}_{ijk})$.

The modified aligned RT is the F-statistic version of the aligned rank test suggested by Mansouri and Govindarajulu (1990). The statistic for the modified aligned RT test is

$$F_m = n\sum_{i=1}^{r}\sum_{j=1}^{c}[\bar{a}_N(\hat{R}_{ij\cdot})-\bar{A}_N]^2/[(r-1)(c-1)MSE(a)],$$

where

$$\bar{a}_N(\hat{R}_{ij\cdot}) = \sum_{k=1}^{n} a_N(\hat{R}_{ijk})/n, \ \ \bar{a}_N = \sum_{\alpha=1}^{N} a_N(\alpha)/N,$$

and MSE(a) is obtained from (2) by replacing X_{ijk} with $a_N(\hat{R}_{ijk})$. It can be shown that under H_0 the limiting distributions of $(r-1)(c-1)F_a$ and $(r-1)(c-1)F_m$ are central χ^2 with $(r-1)(c-1)$ degrees of freedom, Mansouri and Chang (1994).

3 SIMULATION STUDY

In the Monte Carlo simulation studies, we present results from the design with $r = 4$ rows and $c = 3$ columns. Each design is replicated 5000 times to insure the stability of the simulated sampling distributions of the statistics that are considered. The least-squared estimators are considered for $\hat{\alpha}_i$ and $\hat{\beta}_j$ in the aligned RT tests.

Table 1 contains the empirical Type I error rates versus the nominal $\alpha = .05$ for normal underlying distribution with the main effects $\alpha_2 = \beta_1 = e$, and $\alpha_3 = \beta_2 = -e$ and all other effects equal to zero. The Wilcoxon rank score function is used. As in Blair et al. (1987), a severe inflation in the Type I error rates is observed for the RT statistic F_r as e increases. Whereas the Type I error rates of F_a and F_m stay inside the 95% confidence bounds and behave nicely.

The empirical power for these statistics with interaction effects are presented in Table 2. For cases where the empirical Type I error rates are not much larger than the nominal values, we can see that F, F_a, and F_m have better empirical power than F_r.

The Type I error rates and power of these tests are also simulated under different distributions to examine their robustness properties. Table 3 and 4 contains results from exponential (EXP), double exponential (DEX) and Cauchy (CAU) distributions. None of these tests performs well for Cauchy. As both of the main effects increase, the inflation in Type I error rates of the RT statistic is observed in every distribution. However, it performs better than other tests when the underlying distribution is Cauchy. The aligned RT statistic has better performance in general. The other rank tests seem to have higher empirical power, but this is due to their over inflated Type I error rates.

In table 5, the Type I error rates and power of the χ^2 version of the aligned RT tests, $A = (r-1)(c-1)F_a$ and $M = (r-1)(c-1)F_m$, are also presented. Comparisons can be made with the results of the F-statistics in table 3 and 4. The empirical Type I error of the F-statistics converge faster than their χ^2 counterparts. The empirical power of the F-statistic is slightly less than that of the χ^2-statistic.

When the normal scores are used and the underlying distribution is normal, the empirical Type I error and power of the aligned RT tests perform as well as F (Mansouri and Chang 1994). Furthermore, the inflation of the empirical Type I error for the RT test is significantly reduced.

		Sample size (n), $\alpha = .05$			
c	Stat.	5	10	20	50
0.50	F	.050	.053	.051	.052
	F_r	.055	.052	.053	.057
	F_a	.053	.052	.053	.050
	F_m	.056	.058	.058	.055
1.00	F	.048	.047	.047	.047
	F_r	.057	.069	.093	.189
	F_a	.051	.051	.051	.045
	F_m	.055	.056	.056	.051
1.50	F	.047	.050	.054	.053
	F_r	.071	.136	.314	.848
	F_a	.050	.045	.052	.053
	F_m	.056	.050	.055	.058
2.50	F	.047	.051	.049	.056
	F_r	.192	.684	.995	1.000
	F_a	.052	.051	.048	.055
	F_m	.056	.057	.053	.060

Table 1: *Type I Error Rates of Tests for Interactions when $\alpha_2 = \beta_1 = e$, $\alpha_3 = \beta_2 = -e$.*

		Sample size (n), $\alpha = .05$			
c	Stat.	5	10	20	50
0.50	F	.121	.214	.434	.886
	F_r	.111	.195	.392	.845
	F_a	.123	.210	.415	.865
	F_m	.131	.222	.430	.873
1.00	F	.399	.765	.986	1.000
	F_r	.292	.629	.953	1.000
	F_a	.385	.742	.984	1.000
	F_m	.401	.753	.985	1.000
1.50	F	.788	.988	1.000	1.000
	F_r	.521	.898	1.000	1.000
	F_a	.763	.984	1.000	1.000
	F_m	.772	.985	1.000	1.000
2.50	F	.999	1.000	1.000	1.000
	F_r	.764	.994	1.000	1.000
	F_a	.998	1.000	1.000	1.000
	F_m	.998	1.000	1.000	1.000

Table 2: *Power Analysis of Tests for Interactions when $\gamma_{11} = e$, $\gamma_{41} = \beta_1 = -e$, with $\alpha = .05$.*

References

[1] Adichie, J. N. (1978). Rank tests for subhypotheses in general linear regression. *Ann. Statist.*, 6, 1012-26.

[2] Bhapkar, V. P. & Goré, A. P. (1974). A nonparametric test for interaction in two-way layouts. *Sankhya*, Ser. A, 261-72.

[3] Blair, R. C., Sawilowsky, S. S. & Higgins, J. J. (1987). Limitations of the rank transform statistic. *Comm. Statist.*, B 16, 1133-45.

[4] Chiang, C. & Puri, M. L. (1984). Rank procedures for testing subhypotheses in linear regression. *Ann. Inst. Statist. Math.*, 36, A, 35-50.

[5] Conover, W. J. & Iman, R. L. (1981). Rank transformations as a bridge between parametric and nonparametric statistics. *The Amer. Statistician*, 35, 124-128.

[6] Hora S. C. & Conover, W. J. (1984). The F statistic in the two-way layout with rank-score transformed data. *JASA*, 79, 668-75.

[7] Iman, R. L., Hora, S. C. & Conover, W. J. (1984). Comparison of asymptotically distribution-free procedures for the analysis of complete blocks. *J. Amer. Stat. Assoc.*, 79, 674-85.

[8] Mansouri, H. & Chang, G. (1994). A comparative study of some rank tests for interaction. (to appear in Journal of Computational Statistics and Data Analysis.)

[9] Mansouri, H. & Govindarajulu, Z. (1990). A class of rank tests for interaction in two-way layouts. *J. Applied. Stat.*, 17, No. 3, 417-26.

[10] McKean, J. W. & Hettmansperger, T. P. (1976). Test of hypotheses based on ranks in the general linear model. *Comm. Statist.- Theor. Meth.*, A5(8), 693-709.

[11] Mehra, K. L. & Sen, P. K. (1969). On a class of conditionally distribution-free tests for interactions in factorial experiments. *Ann. Math. Statist.*, 40, 658-64.

c	Stat.	EXP	DEX	CAU
0.50	F	.052	.049	.017
	F_r	.086	.048	.053
	F_a	.055	.047	.463
	F_m	.212	.081	.999
1.00	F	.052	.054	.016
	F_r	.399	.113	.059
	F_a	.053	.052	.468
	F_m	.215	.084	1.000
1.50	F	.045	.045	.012
	F_r	.924	.368	.093
	F_a	.052	.049	.458
	F_m	.197	.079	.999
2.50	F	.043	.047	.018
	F_r	1.000	.994	.261
	F_a	.046	.047	.472
	F_m	.198	.073	.999

Table 3: *Type I Error Rates of Tests for Interactions when $\alpha_2 = \beta_1 = e$, $\alpha_3 = \beta_2 = -e$, with $\alpha = .05$, sample size $n = 50$.*

c	Stat.	EXP	DEX	CAU
0.50	F	.449	.540	.018
	F_r	.997	.679	.311
	F_a	.999	.727	.564
	F_m	1.000	.780	.999
1.00	F	1.000	.996	.024
	F_r	1.000	.999	.823
	F_a	1.000	1.000	.806
	F_m	1.000	1.000	1.000
1.50	F	1.000	1.000	.028
	F_r	1.000	1.000	.980
	F_a	1.000	1.000	.954
	F_m	1.000	1.000	1.000

Table 4: *Power analysis of Tests for Interactions when $\gamma_{11} = e$, $\gamma_{41} = \beta_1 = -e$, with $\alpha = .05$, sample size $n = 50$.*

		Sample size (n), $\alpha = .05$			
c	Stat.	5	10	20	50
* Type I Error					
0.50	A	.065	.060	.056	.051
	M	.072	.066	.062	.056
1.00	A	.063	.057	.054	.046
	M	.069	.063	.059	.052
* Power Analysis					
0.50	A	.138	.221	.425	.867
	M	.150	.234	.440	.874
1.00	A	.421	.755	.985	1.000
	M	.440	.770	.985	1.000

Table 5: *Type I Error and Power of R and M.*

[12] Mehra, K. L., & Smith, G. L. E. (1970). On nonparametric estimation and testing for interaction in factorial experiments. *J. Amer. Statist. Assoc.*, 65, 1283-96.

Simultaneous Computation of the Wilcoxon's and Ansari-Bradley's Statistics for Small Samples

CASTAGLIOLA Philippe
Ecole Nationale Supérieure des Techniques Industrielles et des Mines de Nantes.
3, rue Marcel-Sembat 44049 Nantes CEDEX 04.
Tel: +33.40.44.83.75, Fax: +33.40.71.97.40, E-mail: pcasta@auto.emn.fr

Abstract

Wilcoxon's signed rank sum test, Wilcoxon's rank sum test and Ansari-Bradley's rank sum test are three well known distribution-free tests. When the samples size is large enough, the lower tail probabilities $P_0[T_n \leq x]$, $P_0[W_{m,n} \leq x]$ and $P_0[A_{m,n} \leq x]$ may be easily computed, under H_0, using some normal approximations. When the size of the samples is too small these normal approximations become unfortunately insufficient. So the main goal of our work is to find some *fast* algorithms which compute the *exact* lower tail probabilities $P_0[T_n \leq x]$, $P_0[W_{m,n} \leq x]$ and $P_0[A_{m,n} \leq x]$ when the normal approximation is inefficient.

1 Introduction

Wilcoxon's signed rank sum test, Wilcoxon's rank sum test [2] and Ansari-Bradley's rank sum test [1] are three well known distribution-free tests. The two first may be used to investigate the presence of a *shift in location* between two populations, whereas the last one may be used to investigate the presence of a *difference in scale* between two populations having unknown but equal medians. These tests are based respectively on Wilcoxon's T_n statistic, Wilcoxon's $W_{m,n}$ statistic (which is closely related to Mann-Whitney's $U_{m,n}$ statistic [3]), and on Ansari-Bradley's $A_{m,n}$ statistic.

When the sample size is large enough, the lower tail probabilities $P_0[T_n \leq x]$, $P_0[W_{m,n} \leq x]$ and $P_0[A_{m,n} \leq x]$ may be easily computed, under $H_0 : \theta = 0$ (no shift in location) and $H_0 : \gamma^2 = 1$ (no difference in scale), using some normal approximations. For a full description of these approximations see [4] pages 28,68-69,85.

When the size of the samples is too small (i.e. $n \leq 15$ for Wilcoxon's T_n statistic and $m + n \leq 20$ for the others), these normal approximations become unfortunately insufficient. So the main goal of our work is to find some *fast* algorithms which compute the *exact* lower tail probabilities $P_0[T_n \leq x]$, $P_0[W_{m,n} \leq x]$ and $P_0[A_{m,n} \leq x]$ when the normal approximation is inefficient.

2 Wilcoxon's T_n statistic

The T_n statistic can only take values between 0 and $n(n+1)/2$. So we can deduce a first elementary relation

$$P_0[T_n \leq x] = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq \dfrac{n(n+1)}{2} \end{cases}$$

The T_n distribution is, under $H_0 : \theta = 0$, symmetric about $n(n+1)/4$. It is thus possible to compute $P_0[T_n^+ \leq x]$ using a value which is always smaller than $n(n+1)/4$. Therefore, when $x > n(n+1)/4$, we can apply a second elementary relation

$$P_0[T_n \leq x] = 1 - P_0\left[T_n \leq \frac{n(n+1)}{2} - x - 1\right]$$

The lower probability $P_0[T_n \leq x]$ may be computed by counting the number $T_n^{\leq x}$ of k-tuples, $k \in \{0, 1, \ldots, n\}$, among $\{1, 2, \ldots, n\}$ having a sum less or equal to x and by divising it by $C_n^0 + C_n^1 + \cdots + C_n^n = 2^n$

$$P_0[T_n \leq x] = \frac{T_n^{\leq x}}{2^n}$$

Now let us define $T_{n,k}$ to be the set of all the k-tuples among $\{1, 2, \ldots, n\}$ and $T_{n,k}^{\leq x}$ to be the number of k-tuples of $T_{n,k}$ having a sum less or equal to x. With these definitions, it is clear that

$$T_n^{\le x} = \sum_{k=0}^{n} T_{n,k}^{\le x}$$

Let us define $T_{n,k}^{\min}$ and $T_{n,k}^{\max}$ to be respectively the minimal and maximal sums of the elements of all k-tuples of $T_{n,k}$. If we now define $k_1 \ge 0$ to be the largest integer verifying $x \ge T_{n,k_1}^{\max}$ and $k_2 \ge k_1$ to be the smallest integer verifying $x < T_{n,k_2}^{\min}$, then we have

$$T_n^{\le x} = \sum_{k=0}^{k_1} C_n^k + \sum_{k=k_1+1}^{k_2-1} T_{n,k}^{\le x}$$

The values $T_{n,k}^{\min}$ and $T_{n,k}^{\max}$ may be computed using the following recurrent relations

$$\begin{array}{rcl} T_{n,0}^{\min} & = & 0 \\ T_{n,0}^{\max} & = & 0 \\ T_{n,k}^{\min} & = & T_{n,k-1}^{\min} + k \\ T_{n,k}^{\max} & = & T_{n,k-1}^{\max} + n + 1 - k \end{array}$$

We can show that the value $T_{n,k}^{\le x}$ may be computed using the following relation

$$T_{n,k}^{\le x} = \sum_{j=1}^{n-k+1} T_{n-j,k-1}^{\le x-jk}$$

3 Wilcoxon's $W_{m,n}$ statistic

The $W_{m,n}$ statistic can only take values between $m(m+1)/2$ (i.e. $X_1, X_2, \ldots, X_m$ are all smaller than $Y_1, Y_2, \ldots, Y_m$) and $m(m+2n+1)/2$ (i.e. $X_1, X_2, \ldots, X_m$ are all greater than $Y_1, Y_2, \ldots, Y_m$). Thus,

$$P_0\left[W_{m,n} \le x\right] = \begin{cases} 0 & \text{if } x < \dfrac{m(m+1)}{2} \\ 1 & \text{if } x \ge \dfrac{m(m+2n+1)}{2} \end{cases}$$

Under $H_0 : \theta = 0$, $W_{m,n}$ has a symmetric distribution about $m(m+n+1)/2$. Hence we can deduce a relation which is usefull when $2x > m(m+n+1)$

$$P_0\left[W_{m,n} \le x\right] = 1 - P_0\left[W_{m,n} \le m(m+n+1) - x - 1\right]$$

When m and n are exchanged, the distribution of $W_{m,n}$ under $H_0 : \theta = 0$ is just shifted by $n(n+1)/2 - m(m+1)/2$. Then we can always suppose that $m \le n$, and if it is not the case the following rule may be used

$$P_0\left[W_{m,n} \le x\right] = P_0\left[W_{n,m} \le x - \frac{m(m+1)}{2} + \frac{n(n+1)}{2}\right]$$

Let us define $W_{m,n}^{\le x}$ and $W_{m,n}^{=x}$ to be the number of m-tuples among $\{1, 2, \ldots, m+n\}$ having a sum respectively less or equal to x and equal to x. Under $H_0 : \theta = 0$, the lower tail probability $P_0\left[W_{m,n} \le x\right]$ is determined by the ratio

$$P_0\left[W_{m,n} \le x\right] = \frac{W_{m,n}^{\le x}}{C_{m+n}^{m}}$$

Let us also define $W_{m,n,k}$, $k \in \{1, 2, \ldots, n+1\}$ to be the set containing all the m-tuples among $\{1, 2, \ldots, m+n\}$ beginning by k, and $W_{m,n,k}^{\min}$, $W_{m,n,k}^{\max}$ to be respectively the minimal and maximal sums of the elements of all m-tuples of $W_{m,n,k}$. These values may be computed using the following recurrent relations

$$\begin{array}{rcl} W_{m,n,1}^{\min} & = & m(m+1)/2 \\ W_{m,n,1}^{\max} & = & m(m+2n+1)/2 - n \\ W_{m,n,k}^{\min} & = & W_{m,n,k-1}^{\min} + m \\ W_{m,n,k}^{\max} & = & W_{m,n,k-1}^{\max} + 1 \end{array}$$

We also introduce $W_{m,n,k}^{\le x}$ to be number of m-tuples of $W_{m,n,k}$ having a sum less or equal to x. By definition, we have

$$W_{m,n}^{\le x} = \sum_{k=1}^{n+1} W_{m,n,k}^{\le x}$$

By a similar reasonning, if we define $k_1 \ge 1$ to be the largest integer verifying $x \ge W_{m,n,k_1}^{\max}$ and $k_2 > k_1$ to be the smallest integer verifying $x < W_{m,n,k_2}^{\min}$, then we have

$$W_{m,n}^{\le x} = \sum_{k=1}^{k_1} C_{m+n-k}^{m-1} + \sum_{k=k_1+1}^{k_2-1} W_{m,n,k}^{\le x}$$

We can show that $W_{m,n,k}^{\le x}$ may be recursively computed using the simple following relation

$$W_{m,n,k}^{\le x} = W_{m-1,n-k+1}^{\le x-mk}$$

and then

$$W_{m,n}^{\le x} = \sum_{k=1}^{k_1} C_{m+n-k}^{m-1} + \sum_{k=k_1+1}^{k_2-1} W_{m-1,n-k+1}^{\le x-mk}$$

The probability $P_0\left[W_{m,n} = x\right]$ may be computed by counting the number $W^{=x}_{m,n}$ of m-tuples among $\{1, 2, \ldots, m+n\}$ having a sum equal to x and by divising it by C^m_{m+n}

$$P_0\left[W_{m,n} = x\right] = \frac{W^{=x}_{m,n}}{C^m_{m+n}}$$

By a similar reasonning (and with the same notations), we get

$$W^{=x}_{m,n} = \sum_{k=k_1+1}^{k_2-1} W^{=x-mk}_{m-1,n-k+1}$$

4 Ansari-Bradley's $A_{m,n}$ statistic

Let $A'_{m,n}$ and $A''_{m,n}$ be respectively the minimal and maximal values that an $A_{m,n}$ statistic can take, such that

$$P_0\left[A_{m,n} \le x\right] = \begin{cases} 0 & \text{if } x < A'_{m,n} \\ 1 & \text{if } x \ge A''_{m,n} \end{cases}$$

These values depend on the parity of m and n. We can show that if m is even then

$$\begin{aligned} A'_{m,n} &= \frac{m(m+2)}{4} \\ A''_{m,n} &= \frac{m(m+2n+2)}{4} \end{aligned}$$

and if m is odd then

$$\begin{aligned} A'_{m,n} &= \frac{(m+1)^2}{4} \\ A''_{m,n} &= \begin{cases} \dfrac{m(m+2n+2)+1}{4} & \text{if } n \text{ is even} \\ \dfrac{m(m+2n+2)-1}{4} & \text{if } n \text{ is odd} \end{cases} \end{aligned}$$

Under $H_0 : \gamma^2 = 1$, $A_{m,n}$ has a symmetric distribution about its mean $m(m+n+2)/4$ only if $m+n$ is even. In this case we can apply the following relation when $4x > m(m+n+2)$

$$P_0\left[A_{m,n} \le x\right] = 1 - P_0\left[W_{m,n} \le \frac{m(m+n+2)}{2} - x - 1\right]$$

When m and n are exchanged, the distribution of $W_{m,n}$ under $H_0 : \theta = 0$ is shifted and inverted such that

$$P_0\left[A_{m,n} \le x\right] = 1 - P_0\left[A_{n,m} \le A'_{m,n} + A''_{n,m} - x - 1\right]$$

Thus, if $m+n$ is even

$$P_0\left[A_{m,n} \le x\right] = 1 - P_0\left[A_{n,m} \le \tfrac{(m+n)(m+n+2)}{4} - x - 1\right]$$

and if $m+n$ is odd then

$$P_0\left[A_{m,n} \le x\right] = 1 - P_0\left[A_{n,m} \le \frac{(m+n+1)^2}{4} - x - 1\right]$$

We can always suppose that $m \le n$, and if it is not the case we have to apply one of the previous relations.

Let us define $A^{\le x}_{m,n}$ to be the number of m-tuples, defined by Ansari-Bradley's rule, having a sum less or equal to x. Under $H_0 : \gamma^2 = 1$, the lower tail probability $P_0\left[A_{m,n} \le x\right]$ is determined by the following ratio

$$P_0\left[A_{m,n} \le x\right] = \frac{A^{\le x}_{m,n}}{C^m_{m+n}}$$

If we define $s \ge m$ to be

$$s = \begin{cases} \dfrac{(m+n)}{2} & \text{if } m+n \text{ is even} \\ \dfrac{(m+n-1)}{2} & \text{if } m+n \text{ is odd} \end{cases}$$

then we can also define $A_{m,n,k}$, $k \in \{0, 1, \ldots, m\}$ to be the subsets containing all the m-tuples, defined by Ansari-Bradley's rule, having k elements among the s first ones. For each subset $A_{m,n,k}$, we define $A^{\le x}_{m,n,k}$ to be the number of m-tuples of $A_{m,n,k}$ having a sum less or equal to x. Hence we have

$$A^{\le x}_{m,n} = \sum_{k=0}^{m} A^{\le x}_{m,n,k}$$

But for $k = 0$ we have $A^{\le x}_{m,n,0} = W^{\le x}_{m,n-s}$ and for $k = m$ we have $A^{\le x}_{m,n,m} = W^{\le x}_{m,s-m}$. So the previous may be replaced by

$$A^{\le x}_{m,n} = W^{\le x}_{m,n-s} + \sum_{k=1}^{m-1} A^{\le x}_{m,n,k} + W^{\le x}_{m,s-m}$$

The s first elements of subset $A_{m,n,k}$ contain all the k-tuples of k integer among $\{1, 2, \ldots, s\}$. Let $\alpha_{s,k}$ and $\beta_{s,k}$ be respectively the minimal and maximal sums of this s

first elements. We can easily show that these values may be computed using the following recurrent relations

$$\begin{aligned}\alpha_{s,0} &= 0\\ \beta_{s,0} &= 0\\ \alpha_{s,k} &= \alpha_{s,k-1}+k\\ \beta_{s,k} &= \beta_{s,k-1}+s-k+1\end{aligned}$$

If we denote (for clarity) $j_1 = \alpha_{s,k}$ and $j_2 = \beta_{s,k}$, we can show that

$$A^{\leq x}_{m,n,k} = \sum_{j=j_1}^{j_2} W^{=j}_{k,s-k} W^{\leq x-j}_{m-k,n-s+k}$$

and finally

$$\begin{aligned}A^{\leq x}_{m,n} &= W^{\leq x}_{m,n-s}\\ &+ \sum_{k=1}^{m-1}\sum_{j=j_1}^{j_2} W^{=j}_{k,s-k} W^{\leq x-j}_{m-k,n-s+k}\\ &+ W^{\leq x}_{m,s-m}\end{aligned}$$

5 Computer performances and Conclusions

The algorithms for computing the Wilcoxon's and Ansari-Bradley's statistics have been written in C Langage, compiled with the GNU GCC compilator and tested on a SUN-IPC. The average time for computing $P_0\left[T_n \leq x\right]$ is approximately 0.05 seconds for $n = 20$, and the average time for computing $P_0\left[W_{m,n} \leq x\right]$ and $P_0\left[A_{m,n} \leq x\right]$ for each combination of $m+n = 20$ is also approximately 0.05 seconds. These results show clearly that the average response time of the three proposed algorithms is very small, and thus statistical tables become useless.

These paper shows also that the Ansari-Bradley's $A_{m,n}$ statistic may be computed in term of Wilcoxon's $W_{m,n}$ statistic. This reduces the size of the algorithms themselves.

References

[1] Ansari A.R. and Bradley R.A. Rank-sum tests for dispersions. *Ann. Math. Statist.*, 31:1174–1189, 1960.

[2] Wilcoxon F. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.

[3] Mann H.B. and Whitney D.R. On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18:50–60, 1947.

[4] Hollander M. and Wolfe D.A. *Nonparametric statistical methods.* Wiley publications, 1973.

Nonparametric CDF Estimation from Stratified Data

Yuly Koshevnik
Department of Statistical Science
Southern Methodist University
Dallas, TEXAS 75275–0332

Abstract

Nonparametric estimation of functionals is performed in a biased sampling model. Several samples are indepedently drawn from different populations, with constraints imposed on the underlying distributions. A functional of interest is expressed via a certain convex combination of the underlying distributions.

A new estimation procedure is described here. It gives an asymptotically efficient estimate of a quite arbitrary functional. This procedure, properly modificatied, is also relevant to estimate the whole distribution function and various non-linear functionals. As an alternative to the known estimation procedure studied by Gill, Vardi and Wellner (1988), this technique seems to require less computations.

1 Introduction

Various studies conducted in technometrics and econometrics are based on data that emerge from a population with "slowly changing" features. Several mathematical models have been proposed to describe a drifting population via dynamic changes in the underlying distribution. A biased sampling model can be also considered as one of those. The model was initially considered by Vardi (1985) who derived nonparametric maximum likelihood estimates. Later Gill, Vardi, and Wellner (1988) proved asymptotic optimality results for a nonparametric maximum likelihood estimate of a cumulative distribution function (CDF).

Under biased sampling, subsamples are drawn independently, with constraints imposed on the underlying distributions. This paper describes an alternative estimation technique for a functional of interest,

$$\Psi(F) = \int K(x)\, dG(x)$$

represented as an integral with respect to a convex combination, G, of the underlying distributions. The method studied in this paper requires less computations and attains the same asymptotic perfomance level as the one derived in Gill et al. (1988). Applications include a wide variety of more complicated transforms and functionals of G. Once the estimate, $\hat{G}$, is derived for the entire CDF, G, the plug-in rule suggests to use $\hat{\Psi} = \Psi\left(\hat{G}\right)$ as an asymptotically efficient estimate of $\Psi(G)$, for a quite arbitrary transform, $\Psi(G)$. As an illustration, a simple example emerging from econometric studies is considered and numerical results are presented. The algorithm and its modifications are presented. Heuristic considerations are aimed to explain *why* and *how* this technique does work. As to the proofs, they seem to be quite long and technical and will be presented in the paper in preparation (Koshevnik, 1994).

2 Construction of the Estimate

Suppose that s samples (or *strata),*

$$\mathbf{X} = \{X_{ij} : 1 \le i \le n_j;\, 1 \le j \le s\} \tag{1}$$

are collected indepedently, so that the j–th stratum comes out from a distribution F_j, $(1 \le j \le s)$. It is convenient to use another probability mechanism to describe the data generation as follows. Consider a bivariate random variable, (J, X) where J takes values $1, 2, \ldots, s$ with certain probabilities, say $\mathbf{P}(J = j) = \lambda_j$, which add up to 1, i.e. $\sum_{j=1}^{s} \lambda_j = 1$. Each F_j is nothing but the conditional distribution of X, given $J = j$. A functional (or *coparameter*) of interest introduced as a convex combination of integrals,

$$\Psi = \sum_{j=1}^{s} \lambda_j \int K_j(x)\, dF_j(x) \tag{2}$$

is the expectation taken with respect to a joint distribution, P, of the pair (J, X),

$$\Psi = \int K(j, x)\, dP(j, x), \tag{3}$$

where $K(j,x) = K_j(x)$. Constraints on the conditionals $(F_j : 1 \le j \le s)$ are expressed in terms of the marginal distribution, G, of X, i.e.

$$G = \sum_{j=1}^{s} \lambda_j F_j. \tag{4}$$

Under the biased sampling model, these conditionals satisfy an infinite system of equations

$$\frac{dF_j}{dG}(x) = \omega_j W_j(x). \tag{5}$$

and due to this reason, they all can be expressed by means of G. The weight functions, $(W_j : 1 \le j \le s)$ are given, but the weight coefficients, or simply weights, $(\omega_j : 1 \le j \le s)$ can be unknown. If there were no relation between the conditionals, the natural estimate of (3) that replaces G by a similar convex combination of empiricals $(F_j : 1 \le j \le s)$, i.e.

$$\tilde{G} = \sum_{j=1}^{s} \lambda_j \tilde{F}_j \tag{6}$$

would have been impossible to improve. Additional information provided by constraints enables one to adjust the initial estimate by means of the device that performs an orthogonal projection onto some subspace in the space of estimates.

Practically, even the proportions $(\lambda_j : 1 \le j \le s)$ can be unknown, but in this case, it is natural to assume their empirical analogues to be consistent estimates, i.e.

$$\lim \frac{n_j}{N} = \lambda_j, \tag{7}$$

for $1 \le j \le s$. As far as the weights $(\omega_j : 1 \le j \le s)$ are concerned, they also can be either known or unknown. We consider both of the options. If the weights are given, they can be all made equal to 1, since otherwise each function, W_j, will be replaced by $\omega_j W_j$. With the weights unknown, since both G and each F_j are probability measures, the following equations

$$\omega_j = \omega_j(G) = \left(\int W_j(x)\, dG(x)\right)^{-1} \tag{8}$$

hold. This suggests the estimation procedure that approximates weights from the data.

For a coparameter of interest,

$$\Psi(G) = \int K(x)\, dG(x), \tag{9}$$

an alternative representation, via the joint distribution, P, of a pair (J,X) is exploited. Integration over P is nothing but a convex combination of integrals with respect to the conditionals, as in (2), with $K_j = \frac{K}{W_j}$. This suggests an estimate, $\hat{\Psi} = \hat{\Psi}_\beta$, of $\Psi(G)$ defined as

$$\hat{\Psi}_\beta = \sum_{j=1}^{s} \beta_j \int \frac{K}{W_j}\, d\tilde{F}_j, \tag{10}$$

with a vector of unknown coefficients

$$\beta = (\beta_j : 1 \le j \le s)$$

and empirical distributions $\left(\tilde{F}_j : 1 \le j \le s\right)$ replacing the conditionals. If both the weights and proportions are specified, so that every ω_j can be made equal to 1, it will be explained later that the coefficients $(\beta_j : 1 \le j \le s)$ can be selected equal to the corresponding λ's.

Generally, however, there will be an adaptive procedure required to construct the asymptotically efficient estimate. Asymptotically, this estimator differs from the one with the known weights. Asymptotic efficiency for the estimate can be attained in this case due to a two-step procedure. At first, the weights are empirically approximated, vià their consistent (even $\sqrt{N}$-consistent) empirical estimates, and then, with these substitutes, the estimate is produced, as if the unknown weights were equal to their empirical analogues. An explanation of the algorithm is based on the minimization procedure. Among all possible estimates of $\Psi(G)$ as in (9), pick up the one that minimizes the asymptotic variance in (10), i.e.

$$\mathrm{Var}\left(\hat{\Psi}_\beta\right) \longrightarrow \min_{\beta}.$$

This procedure will typically give a vector

$$\beta = (\beta_j = \beta_j(F_1, \ldots, F_s) : 1 \le j \le s),$$

whose components depend on G, or equivalenltly, on all conditionals. Weak convergence results valid for the empirical process defined by the estimate, $\hat{G}$, of the cumulative distribution function, G, are shown in (Koshevnik, 1994) to hold uniformly in $P \in \mathcal{U}$. Here $\mathcal{U}$ is a suitably chosen (small) neighborhood of P, so it becomes possible to replace unknown weights by their $\sqrt{N}$-consistent estimates. The similar idea was successfully implemented in Koshevnik and Levit (1976), to construct an asymptotically efficient estimator for a functional of interest from independent identically distributed data.

3 Algorithm Description

Here we outline the algorithm showing how the asymptotically efficient estimates can be computed. Three options are considered: the first one faces with the case when both weights and proportions are known; the second one is designed for known weights and unknown proportions; and the third one is aimed to estimate a functional of interest under both weights and proportions unknown.

3.1 Known Weights and Proportions

In this case, assume that all $\omega_j = 1$. Since all λ_j are known, the equation

$$G = \sum_{j=1}^{s} \lambda_j F_j$$

contains uncertainty in the distributions only. Therefore, the functional of interest (9) can be expressed as a convex combination of integrals with respect to the distributions F_j, which are constrained by (5). Putting a mass $\lambda_j \left(W_j\left(X_{ji}\right)\right)^{-1}$ into each X_{ji}, and taking the empirical average with respect to this weighted empirical distribution, we obtain the desired estimate of a functional (9). In particular, this method gives an asymptotically efficient estimate of $G(t)$ for any fixed t.

3.2 Known Weights and Unknown Proportions

Once we agree that sample proportions provide consistent estimates of unknown λ_j's, i.e. (7) takes place, the similar estimate can be proposed to estimate a functional (9) in this case. Replacing unknown proportions by their sample analogues and putting the mass

$$\frac{n_j}{N} \left(W_j\left(X_{ji}\right)\right)^{-1}$$

into X_{ji}, we obtain an estimate of G. Averaging with respect to this weighted empirical distribution will give an asymptotically efficient estimate of (9), as before.

3.3 Unknown Weights and Proportions

In this case we can no longer ingnore that the weights are unknown. As far as the proprotions are concerned, their sample versions still are assumed to be consistent estimates. The proposed estimate of (9) is as in (10), with the coefficients $\beta = (\beta_j : 1 \leq j \leq s)$ obtained from the minimization procedure followed by the plug-in rule. Equivalently, the first step of this algorithm defines β as a vector of functionals, with components depending on the conditionals, while the second step replaces these distributions by their empirical versions, each of them based on the corresponding stratum.

4 Asymptotic Efficiency: Outline

The constraints (5) imposed on the distributions can be exploited to derive the information inequalities similar to those presented in Bickel et al. (1993), see also Koshevnik and Levit (1976). To understand why the proposed methods provide asymptotically efficient estimates, consider the case $s = 2$ here. Let F_1 and F_2 be mutually absolutely continuous. Again, $G = \lambda_1 F_1 + \lambda_2 F_2$ is a convex combination of the underlying conditional distributions here. Assume that λ_1 and λ_2 are given and the weights are known. Then, a functional (9) can be written as

$$\frac{\lambda_1}{\omega_1} \int \frac{K}{W_1} dF_1 + \frac{\lambda_2}{\omega_2} \int \frac{K}{W_2} dF_2,$$

so having the weights known, it is easy to find out that the adjustment proposed for this case generally will give just the weighted empirical distribution. With the proportions λ_j's unknown, the similar procudure will be relevant due to a general result in (Koshevnik, 1994). This result implies in particular that the weighted empirical processes

$$\sqrt{N}\left[\sum_{j=1}^{s} \ell_j \left(\tilde{F}_j - F_j\right)\right]$$

converge weakly to certain Gaussian processes, not only for every vector $\ell = (\ell_j : 1 \leq j \leq s)$ satisfying $\sum_{j=1}^{s} \ell_j = 1$, but uniformly in ℓ. Hence, convergence (7) implies that the replacing λ_j by its consistent estimate $\frac{n_j}{N}$ will not cause any change in asymptotic behavior of the estimate.

A slightly more complicated argument can be exploited to prove that in the case of unknown weights ω_j's, the same general result from (Koshevnik, 1994) implies uniform (in $\beta \in B$, where B is a set of vectors β) weak convergence of the weighted empirical distributions. Hence, having replaced the unknown weights by their $\sqrt{N}$-consistent estimates, we will change only the variance-covariance structure of the limiting Gaussian process,

$$\sqrt{N}\left(\hat{G}(t) - G(t)\right)$$

Fortunately, this will be just the same limiting process that appears in Gill, Vardi, and Wellner (1988) to describe both the limit in distribution for the proposed estimate and the lower bounds of asymptotic risks. The reasons of this coincidence are similar to those noticed by Koshevnik and Levit (1976) for the case of homogeneous observations. (See also Koshevnik and Levit (1983) where a more general result is presented.) This fact enables one to almost "automatically" prove asymptotic efficiency of the proposed estimates.

5 Example

The application described here illustrates how the proposed technique works in a relatively simple case, for a combination of continuous and discrete data. Meanwhile, this example also explains why and how the proposed method is relevant for partially observable data, which are subject to either censoring or truncation.

A data set includes three strata. The first stratum, labelled as $j = 0$, is simply a collection of independent observations $(X_{0i} : 1 \leq i \leq n_0)$, with a common cumulative distribution, F. For two given values, d_1 and d_2, the second and third strata contain values

$$(Y_{ji} = I\{X_{ji} \leq d_j\} : 1 \leq i \leq n_j),$$

labelled as $j = 1$ and $j = 2$ respectively. These data came out from econometric studies where several polls were processed. The stratum with $j = 0$ contains complete answers on how much respondents had actually paid for the services provided, while the two remaining strata came from the polls aimed to indicate whether respondents would have agreed to pay a certain amount of money for the same services. Records included indicators of events such as $X \leq d_1$ (the second stratum) or $X \leq d_2$ (the third stratum). These data are obviously incomplete. A coparameter to be estimated includes two components of interest, namely

$$(F(d_1), F(d_2)). \tag{11}$$

An asymptotically efficient estimate of (11) can be found from the competing estimates based on different strata. Notice that each of the cell probabilities in (11) can be estimated by means of either the empirical CDF based on the complete ($j = 0$) stratum or on the corresponding stratum, $j = 1$ or $j = 2$ respectively, of incomplete dichotomous data. Neither of these estimates is efficient generally, but both of them are unbiased. A general method suggest to use an initial estimate such as $\tilde{F}(d_1)$ and $\tilde{F}(d_2)$, i.e. the empirical CDF based on the stratum 0. As a competitor, the proportions $\tilde{p}_j = \frac{Y_j}{n_j}$, can be calculated via Y_1 and Y_2, i.e. simply the sums over the 1–st and the 2–nd strata. Theoretically, these estimates have equal expectations, but to make them equal just means to find an adjusted (or balanced) estimate of (11). The following algorithm was developed to estimate (11) in Koshevnik and Schucany (1994).

1. Calculate empirical proportions as described.
2. The estimate, $\hat{F}(d_1)$, of (11), is proposed as
$$\tilde{F}(d_1) - a_{11}\left(\tilde{F}(d_1) - \tilde{p}_1\right) - a_{12}\left(\tilde{F}(d_2) - \tilde{p}_2\right),$$
and the similar expression can be written for the second component, with the coefficients a_{21} and a_{22} replacing a_{11} and a_{12}, respectively.
3. For each of the components, the variance minimization problem is solved with respect to the coefficients $a_j = (a_{j1}, a_{j2})$. This gives the coefficients depending on the distribution F.
4. The solution $a_{ji}(F)$ is replaced by its empirical estimate, using the empirical function, $\tilde{F}$, which will be $\sqrt{N}$–consistent, provided all three values, i.e. $\frac{n_0}{N}$, $\frac{n_1}{N}$, $\frac{n_2}{N}$, converge to strictly positive limits, as $N = n_0 + n_1 + n_2 \to \infty$.
5. This will yield the asymptotically efficient estimate of (11).

Asymptotic efficiency in this case is implied by the possibility to reduce the problem into a parametric one. In the meantime, the similar adjustment procedure was developed for any functional such as $F(t)$ with the vector of coefficients, $a(t) = (a_1(t), a_2(t))$. Asymptotic efficiency of this general estimate requires some facts regarding empirical processes.

5.1 Numerical Illustration

The sizes coincide for all three strata, i.e.

$$n_0 = n_1 = n_2 = 100,$$

while $\tilde{F}(d_1) = 0.3$, and $\tilde{F}(d_2) = 0.6$. The calculations performed for incomplete strata have indicated $\tilde{p}_1 = 0.35$ and $\tilde{p}_2 = 0.70$.

The coefficients calculated by means of minimization of the variance of a proposed estimate $\hat{F}(d_1)$ (respectively, $\hat{F}(d_2)$) gave

$$a_{11} = 0.481 \text{ and } a_{12} = 0.067$$

for the first component, and

$$a_{21} = 0.154 \text{ and } a_{22} = 0.462$$

for the second one. Under these values of the coefficients, the improved estimate, $\hat{F}(d_1)$ of (11) is equal to

$$0.30 - 0.481 \cdot (0.30 - 0.35) - 0.067 \cdot (0.60 - 0.70) = 0.331$$

and similarly, $\hat{F}(d_2)$ is calculated as

$$0.60 - 0.154 \cdot (0.3 - 0.35) - 0.462 \cdot (0.6 - 0.7) = 0.654.$$

6 Conclusions

The method presented here provides the same asymptotic optimality of estimation as the initial proposal by Vardi (1985). Computationally, it turns out to be easier, for it does not require some auxiliary components to be estimated as precisely as possible. The asymptotic performance attainable by means of this method can hardly guarantee that it works perfectly for any reasonably large data set. Some additonal studies are needed to investigate its features for moderately large samples.

Possible applications include various models with partically observable data, such as censored or truncated observations. Being focused on the asymptotic optimality only, this method provides a unified approach to estimation of various functionals and transforms of the unknown infinite–dimensional parameter. Both asymptotic normality and asymptotic optimality of the proposed estimates are due to the same result, known as empirical central limit theorem and derived uniformly, as the infinite–dimensional parameter, G, in the considered model, runs over a small neighborhood in the set of distributions. The only serious technical limitation of the applicability of the results presented in Koshevnik (1994) is that this neighborhood must be *precompact* with respect to the Kolmogorov distance between two distributions, i.e.

$$\mathrm{d}\ (G_1, G_2) = \sup_{-\infty<t<\infty} |G_1(t) - G_2(t)|.$$

Acknowledgements

I thank my colleagues from the Statistics Department, Southern Methodist University. My special thanks go to Sam Efromovich, Andrzej Kozek, Aad van der Vaart, Peter Bickel, David Pollard, Jon Wellner for helpful conversations.

References

P. J. Bickel, C. A. J. Klaassen, Y. Ritov, J. A. Wellner, *Efficient and Adaptive Estimation for Semiparametric Models,* The Johns Hopkins University Press, 1993.

R. D. Gill, Y. Vardi, J. A. Wellner, "Large Sample Theory of Empirical Distributions in Biased Sampling Models," *Ann. of Statist.*, Vol. 16, pp. 1069–1112, 1988.

Y. Koshevnik, "Efficient Estimation for Restricted Semiparametric Models", Technical Report, SMU/DS/TR–269, 1994.

Y. Koshevnik and B. Levit, "On a Nonparametric Analogue for the Information Matrix", *Theory of Probab. and Its Applications,* Vol. 21, pp. 738–753, 1976.

Y. Koshevnik and B. Levit, "Risk Bounds in Estimation of Symmetrical Distributions", *J. of Soviet Mathematics,* Vol. 21, No. 1, pp. 65–75, 1983.

Y. Koshevnik and W. R. Schucany, "Asymptotically Efficient Nonparametric Estimation with Additional Dichotomous Observations", *J. of Nonparametric Statistics,* submitted (1994).

Y. Vardi, "Empirical Distribtuions in Selection Bias Models", *Ann. of Statist.*, Vol. 10, pp.178–203, 1985.

Approximate Iterated Bootstrap Confidence Intervals

Stephen M.-S. Lee
Department of Statistics
The University of Hong Kong
Pokfulam Road
Hong Kong

G. Alastair Young
Statistical Laboratory
University of Cambridge
16 Mill Lane
Cambridge CB2 1SB, UK

Abstract

Three different approaches to approximation of double bootstrap confidence intervals, each with the aim of improving computational efficiency, are considered. The first replaces the need for a second level of bootstrap sampling by analytic tail area approximations. The second performs the second level of sampling in a sequential manner. The third uses empirical versions of asymptotic expansions for the end points of the double bootstrap confidence interval and for the additive correction to nominal coverage to avoid the need for resampling. The three methods are compared in relation to their respective set-up costs, the improvements in efficiency they yield, the coverage properties of the approximate intervals and the generality with which they may be applied.

1 Introduction

The iterated bootstrap provides a satisfactory theoretical solution to the problem of producing non-parametric confidence intervals with high coverage accuracy, as well as stable lengths and endpoints: see [4] (Section 3.11). An iterated bootstrap confidence interval requires an additive correction to be made to the nominal coverage of an uncorrected interval. This correction will usually be made using a double bootstrap resampling procedure involving two nested levels of Monte Carlo simulation, and is therefore often computationally prohibitively expensive for routine use.

Recently there has been much attention paid to procedures by which the computational demands of the iterated bootstrap confidence interval construction may be reduced. In this paper we consider three different approaches to approximation of iterated bootstrap confidence intervals. The first ([2], [3]) replaces the need for a second level of bootstrap sampling by use of analytic tail area approximations based on saddlepoint methods. The second ([5]) performs the second level of sampling in a sequential manner. The third ([6]) uses empirical versions of asymptotic expansions for the additive correction to nominal coverage and for the end points of the iterated bootstrap intervals to provide two computationally attractive methods of approximation. The first asymptotic interval replaces the need for a second level of bootstrap sampling by a series of simple numerical computations which are readily automated. The second interval requires no resampling. The three approaches are compared in relation to their respective set-up requirements, the improvements in efficiency they yield, the coverage properties of the approximate intervals and the generality with which they may be applied.

Section 2 provides some background and a formal definition of the iterated bootstrap confidence interval. Section 3 discusses the analytic approximation approach. Section 4 presents a discussion of the sequential sampling idea and Section 5 discusses the asymptotic calibration approach. A simulation study involving construction of bootstrap confidence intervals for the population variance, together with general discussion, is presented in Section 6.

2 Iterated Bootstrap Confidence Interval

We will consider the following problem. We wish to construct an accurate bootstrap confidence interval for a scalar parameter θ expressible as a smooth functon of a vector mean: $\theta = g(\mu)$, where

$$\mu = (\mu_1, \ldots, \mu_d) = (E\{f_1(W)\}, \ldots, E\{f_d(W)\}),$$

with $f_1, \ldots, f_d$ smooth, real-valued functions and W denoting a generic random variable with the underlying k-dimensional distribution F. The form of F is unspecified, but our data $\mathcal{X} = (W_1, \ldots, W_n)$ consists of n observations independently drawn from F. Suppose further that θ is estimated by $\hat{\theta} = g(\bar{X})$, where

$$\bar{X} = (\bar{X}_1, \ldots, \bar{X}_d) = n^{-1}\Sigma X_i,$$

with

$$X_i = (X_{1i}, \ldots, X_{di}) = \{f_1(W_i), \ldots, f_d(W_i)\},$$

$i = 1, \ldots, n$.

We shall see later that assumption of such a 'smooth function model' is crucial to the analytic approximation and asymptotic calibration methods, but not to the approach based on sequential sampling.

Let $\mathcal{X}^*$ denote a generic resample – or 'bootstrap sample' – of size n drawn from $\mathcal{X}$, obtained by independently sampling with replacement from $\mathcal{X}$. Denote by $I_0(\alpha; \mathcal{X}, \mathcal{X}^*)$ a bootstrap confidence interval for θ of nominal coverage α. This interval I_0 could, for example, be the percentile method confidence interval, defined below.

The coverage probability of I_0 is

$$\pi(\alpha) = P\{\theta \in I_0(\alpha; \mathcal{X}, \mathcal{X}^*) \mid F\},$$

and in many cases will be significantly different from α.

The interval $I_0(\alpha + t; \mathcal{X}, \mathcal{X}^*)$, where $\pi(\alpha + t) = \alpha$, has coverage exactly equal to the nominal coverage α. Of course, the value of the 'calibration coefficient' t is rarely available. The idea behind the iterated bootstrap in this context is that a bootstrap estimator of t may be constructed using a second level of resampling.

Let $\mathcal{X}^{**}$ denote a generic resample from $\mathcal{X}^*$ and let $I_0(\alpha; \mathcal{X}^*, \mathcal{X}^{**})$ be the version of $I_0(\alpha; \mathcal{X}, \mathcal{X}^*)$ computed using $\mathcal{X}^*$ and $\mathcal{X}^{**}$ instead of $\mathcal{X}$ and $\mathcal{X}^*$, respectively. Then the bootstrap estimate of $\pi(\alpha)$ is

$$\hat{\pi}(\alpha) = P\{\hat{\theta} \in I_0(\alpha; \mathcal{X}^*, \mathcal{X}^{**}) \mid \mathcal{X}\},$$

with the calibration coefficient t being estimated by $\hat{t}$, where

$$\hat{\pi}(\alpha + \hat{t}) = \alpha.$$

The iterated bootstrap confidence interval for θ is then $I_1(\alpha; \mathcal{X}, \mathcal{X}^*) = I_0(\alpha + \hat{t}; \mathcal{X}, \mathcal{X}^*)$.

In practice, the iterated bootstrap confidence interval construction requires Monte Carlo simulation. A finite number B of bootstrap samples, $\mathcal{X}_1^*, \ldots, \mathcal{X}_B^*$, are drawn from $\mathcal{X}$ at an outer level of resampling, and $\hat{\pi}(\alpha)$ estimated by the proportion

$$\text{card}\,\{1 \leq b \leq B : \hat{\theta} \in I_0(\alpha; \mathcal{X}_b^*, \mathcal{X}_b^{**})\}/B.$$

Usually, exact evaluation of I_0 is not feasible, so a second level of C resamples is drawn from $\mathcal{X}_b^*$ to approximate

$$I_0(\alpha; \mathcal{X}_b^*, \mathcal{X}_b^{**}), \quad b = 1, \ldots, B.$$

We see, therefore, that to approximate $\hat{t}$ B resamples must be drawn at an outer level of resampling, and C resamples drawn, for each outer level resample, at the inner level. So a total of $B(C+1)$ bootstrap samples must be drawn to construct the iterated bootstrap confidence interval I_1. Further, both B and C must be large, of the order of 1000s, in order to reduce Monte Carlo simulation error to acceptable proportions and ensure accurate approximation to the theoretical interval. Some means of improving computational efficiency is desirable.

Typically the confidence interval I_0 will be taken as the percentile-method interval. It is noted (see for example [4], Section 3.11.1) that the percentile method yields confidence intervals with stable lengths and endpoints: bootstrap iteration offers the prospect of retaining desirable stability while enhancing coverage accuracy. The percentile method is based on the premise that the sampling distribution of $\hat{\theta}^* = \hat{\theta}(\mathcal{X}^*)$ under sampling from $\mathcal{X}$ should be close to the unconditional distribution of $\hat{\theta}$ under sampling from F.

Define y_β by $P(\hat{\theta} \leq y_\beta \mid F) = \beta$. The bootstrap estimate is $\hat{y}_\beta$, where $P(\hat{\theta}^* \leq \hat{y}_\beta \mid \mathcal{X}) = \beta$. The (theoretical) nominal α-level percentile confidence interval for θ is $I_0 = [\hat{y}_{1-\xi}, \hat{y}_\xi]$, where $\xi = (1 + \alpha)/2$.

For the case of the percentile method interval I_0, the approximation to $\hat{\pi}(\alpha)$ becomes

$$\text{card}\,\{1 \leq b \leq B : 1 - \xi \leq P(\hat{\theta}_b^{**} \leq \hat{\theta} \mid \mathcal{X}_b^*) \leq \xi\}/B,$$

where $\hat{\theta}_b^{**} = \hat{\theta}(\mathcal{X}_b^{**})$ and $\mathcal{X}_b^{**}$ denotes, as before, a generic bootstrap sample drawn from the outer level bootstrap sample $\mathcal{X}_b^*$.

3 Analytic Approximation

DiCiccio, Martin and Young ([2], [3]) consider analytical methods which significantly reduce the computational demands of the iterated bootstrap. Their methods employ saddlepoint approximations to replace the inner level of resampling.

Define $\hat{\theta}^* = g(\bar{X}^*)$ and $\hat{\theta}^{**} = g(\bar{X}^{**})$, where $\bar{X}^*$ and $\bar{X}^{**}$ are versions of $\bar{X}$ computed using $\mathcal{X}^*$ and $\mathcal{X}^{**}$, respectively, in place of $\mathcal{X}$.

The procedure described by DiCiccio, Martin and Young ([2]) is based on estimation of the tail probability $P(\hat{\theta}^{**} \leq \hat{\theta} \mid \mathcal{X}^*)$ through saddlepoint approximation to the joint density of the components $\bar{X}_1^{**}, \ldots, \bar{X}_d^{**}$ of $\bar{X}^{**}$ given $\mathcal{X}^*$, together with application of a tail probability approximation of DiCiccio and Martin ([1]) to the saddlepoint density.

The algorithm used by DiCiccio, Martin and Young ([2]) for construction of an approximate iterated bootstrap confidence interval involves first drawing B resamples $\mathcal{X}_1^*, \ldots, \mathcal{X}_B^*$ from $\mathcal{X}$. For each resample $\mathcal{X}_b^*$ ($b = 1, \ldots, B$), the analytic approximation is used to estimate $P(\hat{\theta}_b^{**} \leq \hat{\theta} \mid \mathcal{X}_b^*)$. DiCiccio, Martin and Young ([2]) suggest choosing several nominal levels $\gamma_1, \gamma_2, \ldots$ close to the desired level α and determining whether the condition

$$\frac{1}{2}(1-\gamma_i) \leq P(\hat{\theta}_b^{**} \leq \hat{\theta} \mid \mathcal{X}_b^*) \leq \frac{1}{2}(1+\gamma_i)$$

is satisfied for each γ_i. Then an estimate of $\hat{\pi}(\gamma_i)$ is the proportion among the B resamples for which the condition holds for the respective γ_i. The desired calibration coefficient $\hat{t}$, which has $\hat{\pi}(\alpha + \hat{t}) = \alpha$, is approximated by interpolation between the $\{\gamma_i, \hat{\pi}(\gamma_i)\}$ pairs. The approximate iterated confidence interval is the percentile method interval of nominal level $\alpha + \hat{t}$ based on the resamples $\mathcal{X}_1^*, \ldots, \mathcal{X}_B^*$.

The key computational requirement of the procedure of DiCiccio, Martin and Young ([2]) is iterative solution of a system of $2d + 1$ non-linear equations in as many unknowns, together with a series of matrix inversions. In practice, for some first-level bootstrap samples the iteration may fail to converge. When this occurs we recommend use of the resampling approach instead. Computational efficiency is determined largely by the frequency with which the iteration fails to converge. DiCiccio, Martin and Young ([2]) give a number of examples of use of their procedure, which demonstrate the value of the approach, both in terms of accuracy and computational efficiency. We restrict attention here to a series of general remarks on this approach.

(1) It is observed that the analytic approximation approach yields confidence intervals with little discernible loss of coverage accuracy over the full-blown iterated resampling intervals constructed using nested levels of resampling.

(2) The advantages of using the methods – which may be a tenfold reduction or more in computation for simple problems – diminishes as the dimensionality d increases, for then the complexity of the iterative procedure increases.

(3) The methods entail some setup costs, in terms of recoding for different problems, and, as already noted, require use of fairly sophisticated packaged numerical routines for root finding etc.

(4) DiCiccio, Martin and Young ([3]) demonstrate how the analytic methods may be modified to make construction of iterated bootstrap confidence intervals by this approach both feasible and computationally worthwhile in more complicated situations. They approximate to the solution of the system of non-linear equations, and so avoid the costly iteration. Use of the resampling alternative to the analytic approach is then never required. The crude methods DiCiccio, Martin and Young ([3]) describe incur some loss of coverage accuracy over the previous analytic approach, but computational savings are substantial. While reliance on sophisticated numerical routines is reduced, setup costs are still substantial.

(5) A weakness of the approach lies in the fact that the analytic methods are restricted in use to the particular smooth function model described in Section 2 above.

(6) For a given problem, the computational advantage of using the analytic approximations of DiCiccio, Martin and Young ([2]) is most substantial for larger sample sizes n, for then the saddlepoint equations are generally easier to solve. However, the bootstrap is most likely to be indicated for use with smaller sample sizes.

(7) Computational speed of the analytic methods of DiCiccio, Martin and Young ([2]) is observed also to depend heavily on the underlying distribution, as the iteration converges in many fewer steps for some data samples than others. Use of the alternative analytic procedure of DiCiccio, Martin and Young ([3]) effectively eliminates the dependence of computational efficiency on n and F.

4 Sequential Sampling

Recall the algorithm for construction of the iterated bootstrap confidence interval. For each $\gamma_i, i = 1, \ldots, l$ ($l = 3$ is sufficient in practice) we wish to know whether the condition

$$\frac{1}{2}(1-\gamma_i) \leq P(\hat{\theta}_b^{**} \leq \hat{\theta} \mid \mathcal{X}_b^*) \leq \frac{1}{2}(1+\gamma_i)$$

is satisfied, for each of B bootstrap samples $\mathcal{X}_1^*, \ldots, \mathcal{X}_b^*$ drawn from $\mathcal{X}$. The value of $p = P(\hat{\theta}_b^{**} \leq \hat{\theta} \mid \mathcal{X}_b^*)$ is not actually required, but instead we wish to know whether the condition

$$\frac{1}{2}(1-\gamma_i) \leq p \leq \frac{1}{2}(1+\gamma_i)$$

is satisfied, $i = 1, \ldots, l$.

Assume $0 < \gamma_1 < \gamma_2 < \cdots < \gamma_l$. Then we wish to test simultaneously a set of nested hypotheses $H_1, \ldots, H_l$, where H_i is the hypothesis that $\frac{1}{2}(1-\gamma_i) \leq p \leq \frac{1}{2}(1+\gamma_i)$.

Lee and Young ([5]) demonstrate how to construct a simultaneous sequential probability ratio test of the l hypotheses. The idea now, therefore, is to use a different number of second level resamples for each first level resample, with the stopping rules of the sequential probability ratio test designed to minimise the (asymptotic) expected number of second level resamples drawn. Details of the computation of the stopping rules are given by Lee and Young ([5]). Since the sequential probability ratio test is being proposed as an alternative to the use of inner level bootstrap sampling with a fixed number C of resamples, the approach is to constrain the error in testing H_j by the sequential approach to be the same as that incurred when testing H_j by a fixed-sample test with sample size C. Computation of the stopping rules then amounts to solving a straightforward constrained optimization problem. The fixed sample size C is used as a terminating upper bound on the sequential stopping time of the simultaneous sequential probability ratio test, so that we are guaranteed to draw fewer second level bootstrap samples than in the standard construction of the iterated bootstrap interval.

By use of the sequential sampling idea we may construct an approximation to the iterated bootstrap confidence interval with considerable computational savings over the standard procedure which draws a fixed number C of second level resamples from each first level resample. By construction, the sequential sampling procedure has (asymptotically) the same error in estimation of $\hat{\pi}(\gamma_i)$ as the standard procedure. Key remarks on the sequential sampling approach are the following:

(1) The resulting intervals display no significant loss of accuracy over the full-blown iterated resampling intervals, by design.

(2) In typical problems, the sequential intervals use only about 10–20% of the computational effort required by the direct approach. Computational savings are therefore competitive with those achieved by analytic methods in moderately complex problems, though less in simple problems.

(3) Computational gains through use of the sequential sampling idea are roughly problem independent and also roughly independent of the sample size n or underlying distribution F, and indeed of the parameter θ being studied. The approach may therefore be used in any new problem of interest, secure in the knowledge that it will yield a definite level of computational saving.

(4) The sequential approach can be used for any parameter θ, not just for the smooth function model described in Section 2.

(5) Setup of the approach is performed just once. The method may then be applied without modification to construct a confidence interval for any parameter θ. No sophisticated numerical procedures are required for implementation.

5 Asymptotic Calibration

For the percentile method confidence interval I_0, the calibration coefficient t satisfies

$$P(\theta \in [\hat{y}_{1-\xi-t/2}, \hat{y}_{\xi+t/2}] \mid F) = \alpha.$$

As noted, t depends on F, which is unspecified, so is unavailable.

The bootstrap version of t is $\hat{t}$ which satisfies

$$P(\hat{\theta} \in [\hat{y}^*_{1-\xi-\hat{t}/2}, \hat{y}^*_{\xi+\hat{t}/2}] \mid \mathcal{X}) = \alpha,$$

where

$$P(\hat{\theta}^{**} \leq \hat{y}^*_\beta \mid \mathcal{X}^*, \mathcal{X}) = \beta.$$

Given $\hat{t}$, the two-sided iterated bootstrap confidence interval of nominal coverage α is

$$I_1(\alpha; \mathcal{X}, \mathcal{X}^*) = [\hat{y}_{1-\xi-\hat{t}/2}, \hat{y}_{\xi+\hat{t}/2}].$$

Using Hall ([4]), we may establish, under mild conditions on F and g, asymptotic expansions for t and y_β:

$$\begin{aligned} t &= 2n^{-1}\pi_1(z_\xi)\phi(z_\xi) + 2n^{-2}\pi_2(z_\xi)\phi(z_\xi) + \ldots \\ y_\beta &= \theta + n^{-1/2}\sigma\{z_\beta + n^{-1/2}p_{11}(z_\beta) \\ &\quad + n^{-1}p_{21}(z_\beta) + \ldots\} \end{aligned}$$

for $0 < \beta < 1$.

In these expansions, the π_j's are odd polynomials, the p_{j1}'s are polynomials of degree at most $j+1$ and are odd for even j and even for odd j, σ^2 is the asymptotic variance of $n^{1/2}(\hat{\theta} - \theta)$, ϕ is the $N(0,1)$ density and $z_\beta = \Phi^{-1}(\beta)$.

In any given example these expansions are extremely complicated. However, since σ^2 and the coefficients of the polynomials depend only on moments of F, we may easily establish the corresponding expansions for the bootstrap versions $\hat{t}$ and $\hat{y}_\beta$ of t, y_β respectively:

$$\begin{aligned} \hat{t} &= 2n^{-1}\hat{\pi}_1(z_\xi)\phi(z_\xi) \\ &\quad + 2n^{-2}\hat{\pi}_2(z_\xi)\phi(z_\xi) + \ldots \\ \hat{y}_\beta &= \hat{\theta} + n^{-1/2}\hat{\sigma}\{z_\beta \\ &\quad + n^{-1/2}\hat{p}_{11}(z_\beta) + n^{-1}\hat{p}_{21}(z_\beta) + \ldots\}. \end{aligned}$$

Here $\hat{\pi}_j$, $\hat{p}_{j1}$ and $\hat{\sigma}^2$ are obtained by substituting sample moments for population moments in the expressions for π_j, p_{j1}, σ^2 respectively.

Define

$$\tilde{\imath} = 2n^{-1}\hat{\pi}_1(z_\xi)\phi(z_\xi)$$

and

$$\begin{aligned}\tilde{y}_\beta &= \hat{\theta} + n^{-1/2}\hat{\sigma}\{z_\beta \\ &\quad + n^{-1/2}\hat{p}_{11}(z_\beta) + n^{-1}\hat{p}_{21}(z_\beta)\}.\end{aligned}$$

Note that both these quantities may be calculated directly from sample moments: no Monte Carlo approximation is required.

We therefore arrive at two possible sample-based asymptotic approximations to the iterated bootstrap confidence interval:

$$\begin{aligned}I_2 &= [\hat{y}_{1-\xi-\tilde{\imath}/2}, \hat{y}_{\xi+\tilde{\imath}/2}],\\ I_3 &= [\tilde{y}_{1-\xi-\tilde{\imath}/2}, \tilde{y}_{\xi+\tilde{\imath}/2}].\end{aligned}$$

Key comments on these intervals, introduced by Lee and Young ([6]), are:

(1) The interval I_2 still involves sample quantities $\hat{y}_\beta$, to be approximated by one level of bootstrap resampling. But the inner level of sampling is avoided by use of $\tilde{\imath}$, computed directly without sampling.

(2) In principle, the interval I_3 requires no resampling at all. The procedure might, however, occasionally require some form of adjustment, if, for example, $\alpha + \tilde{\imath} \notin (0,1)$ or $\tilde{y}_{1-\xi-\tilde{\imath}/2} \geq \tilde{y}_{\xi+\tilde{\imath}/2}$. In this case we suggest using I_2 instead.

(3) Construction of the intervals I_2 and I_3 is easily packaged. The required computation requires to be coded just once, for the general case. Application then requires only specification of the formula g for the parameter θ of interest. The basis of an automatic packaging is use of techniques of exact numerical derivative evaluation. For details, see Lee and Young ([6]). In particular, no symbolic computation is required for practical use.

(4) Through study of a range of problems, it would appear that asymptotic calibration gives coverage correction comparable to the analytic and sequential approaches, and the full-blown iterated bootstrap, at a fraction of the computational cost. An indication of the levels of computational saving is given in the example of Section 6 below.

(5) The asymptotic calibration requires purely arithmetic computation, and computational savings are therefore independent of sample size or underlying distribution.

(6) Use of asymptotic calibration is, however, restricted, as with the analytic approach, to the smooth function model.

6 Simulation Study

A simulation study has been carried out on the variance example studied by Schenker ([7]) and DiCiccio, Martin and Young ([2]). The parameter of interest θ is the population variance and its estimate $\hat{\theta}$ is the (biased) sample variance. The study compared the coverage accuracy of the (uncorrected) percentile confidence interval I_0 with the full-blown iterated bootstrap interval I_1, approximated using two nested levels of resampling. Also compared were the asymptotic intervals I_2 and I_3, the sequential interval I_s of Lee and Young ([5]) and the two approximate intervals I_{A1} and I_{A2} described by DiCiccio, Martin and Young ([2]) and DiCiccio, Martin and Young ([3]) respectively.

Four different underlying distributions with various degrees of skewness and kurtosis were used: the standard normal $N(0,1)$, with no skewness and no kurtosis, the folded normal $|N(0,1)|$, with high skewness and low kurtosis, the double exponential of unit rate with no skewness and high kurtosis, and finally, the log normal, $\exp(N(0,1))$, which has high skewness and high kurtosis. The variances are respectively 1, $1-2/\pi$, 2 and $e(e-1)$. Three different sample sizes were taken: $n = 20$, 35 and 100 respectively. The full-blown iterated interval I was not constructed for $n = 100$ due to its immense computational demands in this case.

The coverage probabilities of the various confidence intervals were approximated from 1600 random samples, so that each coverage figure has a standard error of approximately 0.01. Intervals I_0, I_2, I_{A1} and I_{A2} were constructed using $B = 1000$ bootstrap resamples. The full-blown iterated interval I_1 was constructed using $C = 1000$ inner level bootstrap samples. The sequential interval I_s was constructed using $B = 1000$ outer level bootstrap samples: the inner level of sampling was performed sequentially, subject to an upper limit of $C = 1000$ bootstrap samples being drawn from any given outer level sample. The analytic interval I_{A1} generally requires no inner level resampling. However, occasionally the iteration required by the analytic approximation failed to converge. In these circumstances the interval was constructed by the resampling method, using $C = 1000$ inner level resamples. The interval I_3 generally requires no resampling. However, in the case of erratic asymptotic interval end-points where, for example, the lower limit exceeds the upper limit, the interval I_3 was replaced by I_2.

Table 1: Estimated coverage probabilities for variance, based on 1,600 random samples of sizes $n = 20$, 35 and 100 drawn from each of four different distributions. I_1 is full-blown interval, I_0 is uncorrected percentile interval. I_S is sequential interval, I_2 and I_3 are asymptotic intervals, I_{A1} and I_{A2} are saddlepoint-based analytic intervals.

Normal data $N(0, 1)$ (no skew, no kurtosis)

Interval	coverage, $n = 20$	coverage, $n = 35$	coverage, $n = 100$
I_2	0.833	0.854	0.883
I_3	0.832 (0.161)	0.853 (0.014)	0.884 (0.000)
I_0	0.727	0.793	0.857
I_1	0.848	0.859	—
I_S	0.829 (190.2)	0.851 (166.8)	0.883 (143.1)
I_{A1}	0.820 (0.001)	0.843 (0.000)	0.879 (0.000)
I_{A2}	0.803	0.829	0.873

Folded normal data $|N(0, 1)|$ (high skew, low kurtosis)

Interval	coverage, $n = 20$	coverage, $n = 35$	coverage, $n = 100$
I_2	0.803	0.821	0.874
I_3	0.800 (0.285)	0.819 (0.101)	0.880 (0.003)
I_0	0.686	0.753	0.843
I_1	0.815	0.834	—
I_S	0.793 (195.4)	0.823 (176.6)	0.876 (151.6)
I_{A1}	0.792 (0.024)	0.815 (0.003)	0.873 (0.002)
I_{A2}	0.778	0.798	0.860

Double exponential data ($\frac{1}{2}\exp(-|x|)$ (no skew, high kurtosis)

Interval	coverage, $n = 20$	coverage, $n = 35$	coverage, $n = 100$
I_2	0.811	0.846	0.869
I_3	0.809 (0.304)	0.848 (0.118)	0.872 (0.013)
I_0	0.698	0.776	0.834
I_1	0.826	0.854	—
I_S	0.796 (202.4)	0.844 (182.4)	0.871 (157.1)
I_{A1}	0.803 (0.026)	0.840 (0.002)	0.869 (0.000)
I_{A2}	0.783	0.817	0.850

Log normal data $\exp\{N(0, 1)\}$ (high skew, high kurtosis)

Interval	coverage, $n = 20$	coverage, $n = 35$	coverage, $n = 100$
I_2	0.526	0.602	0.696
I_3	0.526 (0.533)	0.602 (0.393)	0.696 (0.216)
I_0	0.416	0.504	0.608
I_1	0.544	0.630	—
I_S	0.513 (218.7)	0.589 (207.7)	0.706 (190.6)
I_{A1}	0.529 (0.117)	0.610 (0.059)	0.699 (0.007)
I_{A2}	0.519	0.591	0.663

Table 2: Theoretical leading terms in asymptotic expansions of calibrating coefficient and coverage error corresponding to the standard iterated bootstrap confidence interval I_1.

True distribution	Calibrating coefficient, t	Coverage error, $\mathrm{P}(\theta \in I_1) - \alpha$
Standard normal, $N(0,1)$	$3.109\, n^{-1}$	$-1.499 \times 10^2\, n^{-2}$
Folded normal, $\lvert N(0,1) \rvert$	$6.498\, n^{-1}$	$-1.370 \times 10^3\, n^{-2}$
Double exponential, $\exp(-\lvert x \rvert)/2$	$1.206 \times 10\, n^{-1}$	$-1.240 \times 10^4\, n^{-2}$
Log normal, $\exp(N(0,1))$	$1.411 \times 10^6\, n^{-1}$	$-2.488 \times 10^{20}\, n^{-2}$

The simulation results are reported in Table 1. For the interval I_3, the proportion of simulations for which the end-points were erratic is given in parentheses. This proportion, though substantial for $n = 20$, diminishes to negligible values for larger sample sizes, except for the log normal case. For the interval I_{A1}, the proportion of occasions when resampling was used instead of the analytic approximation is given in parentheses: this proportion also diminishes rapidly with n. The figure in parentheses after each coverage value for the sequential interval is the average number of second level bootstrap samples drawn, to be compared with the fixed number $C = 1000$ of the conventional interval I_1.

The results show very clearly the effect of iteration on the coverage accuracy of the intervals. Overall, the full-blown interval I_1 offers the best coverage accuracy, though all the approximate intervals considered offer reasonable approximations, in terms of coverage accuracy, to that interval. As expected, the crude analytic interval I_{A2} displays discernibly poorer coverage accuracy than the interval I_{A1} it directly approximates.

Considering the sequential interval I_s, we note the requirement of slightly fewer inner level resamples as the sample size n increases. Also, the number of sequential resamples depends slightly on the underlying distribution. Nevertheless, we observe that the computational savings due to drawing the inner level resamples sequentially are not much affected by the underlying distribution, compared to the other intervals considered.

Without giving full timing comparisons, we note that the computational savings through use of the asymptotic interval I_3 depend on the proportion of times that adjustment of that interval is required. Relative to I_2, the interval I_3 is most computationally advantageous for larger n and normal-type underlying populations. Use of I_3 can reduce computation relative to I_2 by as little as a factor of 2, for $n = 20$ in the log-normal case, or as much as 150 or so, for $n = 100$ and a normal distribution. Compared to the sequential interval I_s, use of I_3 reduces computation by a factor of at least 250 for $n = 20$: for $n = 100$ this factor increases dramatically to around 15000, except for the log-normal case where the factor remains of the order 400. The sequential interval I_s requires about 3 times the amount of computation of the analytic interval I_{A2}, uniformly over the cases considered in the simulation. As we have previously noted we might expect, the computational savings through use of the analytic interval I_{A1} are very variable. Relative to I_s, which we have already noted provides fairly uniform savings, requiring about 1/5 of the computation of the full-blown interval I_1, I_{A1} can vary from requiring about twice as much computation to requiring only about 1/3 as much computation, depending on the sample size and underlying distribution.

It is to be noted that, even with iteration, the coverage error is often very large, especially for the log normal underlying distribution. To illustrate further the impact on coverage error of different distributions, we have computed the theoretical leading terms of the expansions of the calibrating coefficient t and of the coverage error, for the theoretical iterated bootstrap confidence interval I_1. Note that all the iterated intervals considered here have coverage error of order $O(n^{-2})$, while the uncorrected interval I_0 has coverage error of order $O(n^{-1})$. Results are listed in Table 2. We can readily appreciate why the log normal distribution yields large coverage error, and why the bootstrap iteration has relatively little success in eliminating coverage error in this case.

In terms of coverage accuracy, the asymptotic calibration proves very effective, and is also by far the best of the intervals considered in terms of computational speed. The interval I_3 generally provides worthwhile computational savings over I_2. The interval I_3 is perhaps, therefore, to be favoured overall. In the variance example considered here, use of the asymptotic interval I_3 reduces computation by a factor of 1000s, compared to I_1, whatever the sample size or parent population.

Acknowledgements

We are grateful to Tom DiCiccio and Michael Martin for their valuable input to the ideas presented here.

References

[1] DiCiccio, T.J. and Martin, M.A. (1991) Approximations of marginal tail probabilities for a class of smooth functions with applications to Bayesian and conditional inference. *Biometrika*, **78**, 891–902.

[2] DiCiccio, T.J., Martin, M.A. and Young, G.A. (1992a) Analytical approximations for iterated bootstrap confidence intervals. *Statistics and Computing*, **2**, 161–171.

[3] DiCiccio, T.J., Martin, M.A. and Young, G.A. (1992b) Fast and accurate approximate double bootstrap confidence intervals. *Biometrika*, **79**, 285–295.

[4] Hall, P. (1992) *The Bootstrap and Edgeworth Expansion*. Springer: New York.

[5] Lee, S.M.-S. and Young, G.A. (1993a) Sequential iterated bootstrap confidence intervals. Research Report 93–17, Statistical Laboratory, University of Cambridge.

[6] Lee, S.M.-S. and Young, G.A. (1993b) Asymptotic iterated bootstrap confidence intervals. Research Report 93–18, Statistical Laboratory, University of Cambridge.

[7] Schenker, N. (1985) Qualms about bootstrap confidence intervals. *J. Amer. Statist. Assoc.*, **80**, 360–361.

Tail-specific Linear Approximations for Efficient Bootstrap Simulations

Tim C. Hesterberg
Mathematics Department
Franklin & Marshall College
Lancaster, PA 17604-3003
T_Hesterberg@FandM.edu

Abstract

Two effective variance reduction techniques for estimating probabilities and quantiles in the tails of bootstrap distributions — importance sampling and concomitants of order statistics — are based on linear approximations. Although these techniques offer potential variance reductions by factors from nine to infinity, in practice the reductions may be only by a factor of two or smaller, because of inaccurate linear approximations.

We develop tail-specific linear approximations that are more accurate where the accuracy is important, in the tails of distributions. Our methods fall into two categories – influence function methods and regression methods. Both can be applied without problem-specific analytical calculations, and both have tail-specific versions.

We apply the tail-specific approximations to importance sampling and concomitants, and propose another technique that uses linear approximations, post-stratification implemented using the saddlepoint. This technique shares the same $O(n^{-1/2}B^{-1})$ variance as the concomitants procedure.

Keywords: Concomitants of order statistics, stratified sampling, empirical influence function, importance sampling, jackknife, variance reduction.

1 Introduction

This article concerns more efficient computational methods for estimating tail probabilities and percentiles of bootstrap distributions. The primary focus of this article is the development of tail-specific linear approximations for bootstrap statistics. Such approximations do not stand on their own, but allow more effective use of other methods such as importance sampling (Johns 1988, Davison 1988) and concomitants of order statistics (Efron 1990 section 5, Do and Hall 1992). We also propose another method, post-stratification using the saddlepoint.

We concentrate on the nonparametric bootstrap; see e.g. Efron (1982, 1987), Efron and Tibshirani (1993) for further discussion. The original data is $\mathcal{X} = (x_1, x_2, \ldots, x_n)$, a sample from an unknown distribution (which may be multivariate). Let $\mathcal{X}^* = (X_1^*, X_2^*, \ldots, X_n^*)$ be a "bootstrap" sample of size n chosen with replacement from $\mathcal{X}$. We wish to estimate tail probabilities or quantiles for $T^* = T(\mathcal{X}^*)$, which may be a parameter estimate or a pivotal statistic used for inferences.

Let $G(a) = P\{T^* \leq a\}$ be the bootstrap distribution function. The simple Monte Carlo estimate of G requires some large number B of bootstrap samples samples $\mathcal{X}_b^*$ for $b = 1, \ldots, B$, then the estimate is $\hat{G}(a) = (1/B)\sum_{b=1}^{B} I(T_b^* \leq a)$, where I is the usual indicator function and $T_b^* = T(\mathcal{X}_b^*)$ for each such sample. Furthermore, some techniques such as the "iterated bootstrap" (Beran 1987) require that some number B_2 of bootstrap samples be generated from each of the original B bootstrap samples, requiring a total of $B + BB_2$ bootstrap samples. For techniques that require tail probability or quantile estimates the total number of bootstrap samples required can be quite large. For example, Efron (1987) finds that $B = 1000$ observations are required to adequately estimate tail quantiles for his (non-iterated) confidence intervals, and Booth and Hall (1992) find that $B_2 = KB^{1/2}$ is approximately optimal for some constant K that depends on the desired coverage level, with $K \doteq 0.57$ for a two-sided 95% confidence interval; this results in a total of about 19,000 bootstrap replications.

The three variance reduction techniques (importance sampling, concomitants, and post-stratification) can reduce the computational burden substantially, but all require accurate linear approximations to T^* in order to work well. For example, Do and Hall (1992) show that the concomitants procedure gives variance reductions that approach infinity, asymptotically, because their linear approximations become more accurate as n increases. But they note that the procedure does not do well when a statistic is markedly non-linear. Similarly, Efron (1990) reports variance reductions by factors of only 1.8 for the lower 2.5%-tile and 0.49 (half as efficient as simple random sampling) for the upper 97.5%-tile of the law school data of Efron (1992, section 2.5). Do and Hall (1992) further note that the procedure does better in the center of a distribution than in the tails. We show that the usual linear approximations are more accurate in the center of a bootstrap distribution than in the tails.

Our primary result in this article is the development of tail-specific (actually quantile-specific) linear approximations, which are more accurate near their design quantiles. Such "local" accuracy is more important to the performance of variance reduction techniques than is overall accuracy.

We begin with a general discussion of linear approximations in section 2. We discuss linear approximations related to the empirical influence function (Efron 1982) in section 3, and approximations based on regression in section 4. In section 5 we return a topic raised in section 2, that transforming T^* may improve linearity, and discuss how to choose a transformation. In section 6 we review the variance reduction techniques which can use the linear approximations, and show how sensitive these techniques are to the quality of the linear approximations. We propose a new technique, post-stratification using the saddlepoint probability estimate.

2 Linear approximations

A "curvilinear approximation" to T^* is determined by a vector $\mathbf{L}$ of length n, with elements L_j corresponding to each of the original observations x_j, such that

$$\psi(T(\mathcal{X}^*)) \doteq \sum_{j=1}^{n} L_j P_j^* \tag{2.1}$$

where ψ is a smooth monotone increasing function, $P_j^* = M_j/n$, and M_j is the number of times x_j is included in $\mathcal{X}^*$. For later use, define L^* to be the right hand side of (2.1), and let $L^{*(\alpha)}$, $T^{*(\alpha)}$, and z_α be the true α quantiles of the distributions of L^*, T^*, and the standard normal distribution, respectively.

For example, consider the usual t-statistic

$$T(\mathcal{X}^*) = n^{1/2}(\bar{X}^* - \bar{x})/s_{\mathcal{X}^*}, \tag{2.2}$$

where $\bar{X}^*$ is the sample average and s^2 is the sample variance of a bootstrap sample and $\bar{x}$ is the sample average of the original data. We use as $\mathcal{X}$ the data of Graham et al. (1990): (9.6, 10.4, 13.0, 15.0, 16.6, 17.2, 17.3, 21.8, 24.0, 26.9, 33.8), for which $\bar{x} \doteq 18.7$. Fixing the denominator of (2.2) at 7.3, the sample standard deviation of the original sample, results in a "central" approximation with $L_{\text{central},j} = 11^{1/2}(x_j - 18.7)/7.3$ and $L^*_{\text{central}} = 11^{1/2}(\bar{X}^* - 18.7)/7.3$. The first panel of Figure 1 shows a scatterplot of T^* vs. this L^*, for 1500 bootstrap samples. The approximation is very accurate near $L^* = 0$, with very little scatter either above or below the line $T^* = L^*$, but is worse for L^* farther from zero.

The increasing conditional variability of T^* for L^* farther from 0 motivates the central theme of this article. We call the first linear approximation a central approximation because it is accurate in the center of the bootstrap distribution. Some other linear approximation may be more accurate elsewhere. Indeed, the approximation defined by $L_{\text{right},j} = -15.4 + 1.02x_j - 0.014x_j^2$ is more accurate in the right tail, as shown in the second panel of Figure 1. We obtain this approximation using the right-tail influence function method in the next section; the central approximation is also equivalent to the central influence function approximation.

Furthermore, the relationship between T^* and L^* is nonlinear, and both versions of L^* are better approximations to some transformation $\psi(T^*)$ than they are to T^* itself. We discuss estimation of ψ in section 5, and applications of that estimate; but estimating ψ requires a linear approximation, and that is where we turn now.

3 Influence Function and Knife Approximations

We begin in this section by describing statistics T for which the linear approximation methods in this section are defined, then proceed to describe the general class of approximations and the specific approximations. The approximations differ in two regards — whether they are central or tail-specific approximations, and in the choice of one parameter, which in turn determines whether the approximation is implemented using analytically or numerically, and also determines whether the approximation is suitable for non-smooth or only smooth functions.

We begin by writing $T(\mathbf{P}^*) \equiv T(\mathcal{X}^*)$, where $\mathbf{P}^* = (P_1^*, \ldots, P_n^*)$; in other words, any bootstrap sample may be viewed as an empirical distribution with weight P_j^* on original observation x_j. In this section we require that T be defined for all weight vectors $\mathbf{P}$ with nonnegative weights summing to 1, not just those which are realizable as bootstrap samples, i.e. those whose coordinates are integers/n. We say that such T are defined for weighted samples.

For example, we may rewrite the t-statistic (2.2) as

$$T(\mathbf{P}) = (n-1)^{1/2}(\bar{x}_{\mathbf{P}} - \bar{x})/\hat{\sigma}_{\mathbf{P}}$$

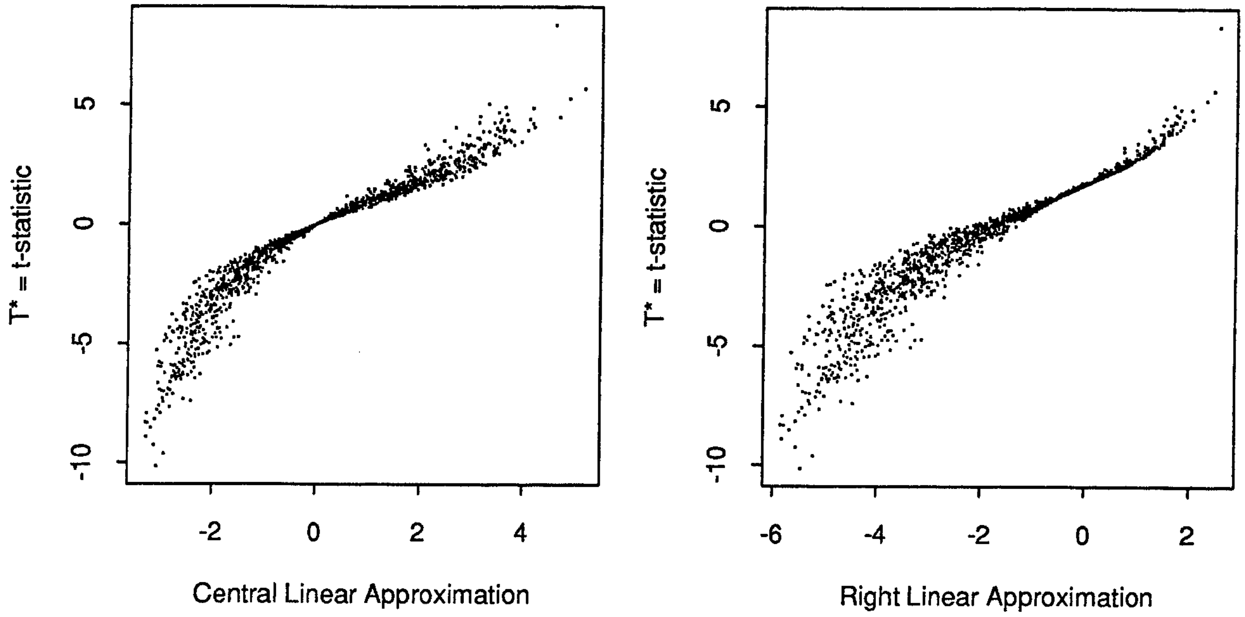


Figure 1: Central and right-tail influence function linear approximations for the t-statistic.

where $\bar{x}_{\mathbf{P}} = \sum_{j=1}^{n} P_j x_j$ is the weighted average and $\hat{\sigma}^2_{\mathbf{P}} = \sum_{j=1}^{n} P_j (x_j - \bar{x}_{\mathbf{P}})^2$ is the weighted standard deviation of a sample. Other examples of statistics which are defined for weighted samples include functions of sample moments (such as means, variances, regression coefficients, and the usual (Pearson) bivariate correlation) and statistics defined by estimating equations (such as M-estimates of location). Functions such as Spearman's correlation, which is a function of the ranks of observations, are not defined for weighted samples, and the regression methods in section 4 should be used for such functions.

The linear approximations in this section are of the form

$$L_j = \frac{T(\mathbf{P}_c + \epsilon(\delta_j - \mathbf{P}_c)) - T(\mathbf{P}_c)}{\epsilon} \tag{3.1}$$

for some point $\mathbf{P}_c$ and some ϵ. These are Taylor-series or finite-difference approximations to the gradient of the function $T(\mathbf{P})$; the approximation differ in the choice of $\mathbf{P}_c$ and the choice of ϵ.

Central linear approximations use $\mathbf{P}_c = \mathbf{P}_0 = (1, 1, \ldots)/n$, which corresponds to the original data, and one of four choices of ϵ:

$$\begin{array}{lll} \mathbf{L}_{\text{negative jackknife}} & : & \epsilon = -1/(n-1) \\ \mathbf{L}_{\text{influence function}} & : & \epsilon \to 0 \\ \mathbf{L}_{\text{positive jackknife}} & : & \epsilon = 1/(n+1) \\ \mathbf{L}_{\text{butcher knife}} & : & \epsilon = n^{-1/2} \end{array} \tag{3.2}$$

The first three are the negative jackknife, influence function (or infinitesimal jackknife), and positive jackknife approximations of Efron (1982). The butcher knife (large jackknife) is motivated by the observation of Efron (1982) that the jackknife uses T evaluated at points which very close to $\mathbf{P}_0$, with a squared distance of $|\mathbf{P} - \mathbf{P}_0|^2 = 1/(n(n-1))$, whereas $E[|\mathbf{P} - \mathbf{P}_0|^2] = (n-1)/n^2$ under simple random (bootstrap) sampling. The butcher knife matches the expected squared distance, and so may be a more accurate approximation to the bootstrap distribution.

3.1 Tail-specific methods

Tail-specific linear approximations are also defined using (3.1), using the same choices of ϵ (3.2), but with the Taylor-series or finite-difference approximation performed about a different initial point $\mathbf{P}_c = \mathbf{P}_\alpha$. We choose $\mathbf{P}_\alpha$ so that $T(\mathbf{P}_\alpha) \doteq T^{*(\alpha)}$ but that otherwise $\mathbf{P}_\alpha$ is as close to $\mathbf{P}_0$ as possible.

If n is very large, a suitable initial point is

$$\mathbf{P}_\alpha = \mathbf{P}_0 + c\mathbf{L}_{\text{central}}, \tag{3.3}$$

where $\mathbf{L}_{\text{central}}$ is a central linear approximation (normalized to sum to 0) and $c = z_\alpha n^{-1} (\sum_{j=1}^{n} L^2_{\text{central},j})^{-1/2}$.

If n is not large, or if $\mathbf{L}_{\text{central}}$ is skewed, we recommend instead to use exponential tilting,

$$\mathbf{P}_\alpha = \mathbf{P}_{[\tau]}, \text{ where } P_{j,[\tau]} = k\exp(\tau L_{\text{central},j}), \qquad (3.4)$$

and where k is a normalizing constant and τ solves $\mathbf{L}_{\text{central}} \cdot \mathbf{P}_\alpha = L^{*(\alpha)}$, with $L^{*(\alpha)}$ estimated from a normal, Cornish-Fisher, or saddlepoint approximation. The right panel of Figure 1 shows the right-tail ($\alpha = 0.975$) influence function estimate, using exponential tilting.

3.2 Comparing influence function and knife methods

The four choices of ϵ in (3.2) result in approximations which differ in implementation details, in the kind of problem where they may be used, and in accuracy.

The knife approximations are evaluated numerically. They have the advantage that they do not require analytical calculations, but the disadvantage of requiring n evaluations of T.

The influence function $\mathbf{L}$ is the gradient of the function $T(\mathbf{P})$ at $\mathbf{P}_c$, and is only suitable for statistics which are smooth (continuous and differentiable) functions of $\mathbf{P}$; similarly for the jackknife versions, which are finite-difference approximations to the gradient. For a discontinuous function such as the sample median the influence function estimate is undefined if n is even and has $L_j = 0$ for all j if n is odd; the jackknife methods may have $L_j = 0$ for all j if there are repeated observations at the median. The butcher knife is a finite-difference method, but evaluated at points farther from $\mathbf{P}_c$, and can be used for non-smooth functions.

For statistics which are smooth functions of $\mathbf{P}$ there are subtle but significant differences between the four methods. The correlations between T^* and the negative jackknife, influence function, positive jackknife, and butcher knife linear approximations, respectively, are 0.942, 0.955, 0.957, and 0.955, for our t-statistic example. The correlations between a nonparametric estimate $\hat{\psi}(T^*)$ and the approximations are higher (correlation .987 with the influence function approximation), but follow a similar pattern — only the negative jackknife does appreciably worse than the others. But even though the butcher knife approximation is as good overall as the influence function approximation, it is not quite as good "locally", for values of L^* near targets $\mathbf{L} \cdot \mathbf{P}_c$. This difference arises from the way the approximations are defined — the influence function is determined by a local approximation to T, the butcher knife by a more global approximation. Local accuracy (in the tails) is more important for the variance reduction techniques than is global accuracy, so we recommend the influence function (or positive jackknife approximation) for smooth statistics.

4 Regression approximations

Linear regression methods may be used to obtain linear approximations for any statistic, even those not defined for weighted samples. We begin with central regression methods, and follow with tail-specific methods.

Let $M_{b,j}$ be the number of times original observation x_j is included in the b'th sample and let $P^*_{b,j} = M_{b,j}/n$. Run a linear regression without an intercept of the form

$$T^*_b = \sum_{j=1}^{n} \beta_j P^*_{b,j} + \text{residual}_b, \qquad (4.1)$$

and let

$$L_{\text{central},j} = \beta_j - \bar{\beta} \qquad (4.2)$$

where $\bar{\beta} = (1/n)\sum_{i=1}^{n} \beta_i$. The intercept is omitted because otherwise the regression would be singular. This linear approximation was obtained by Efron (1990).

Do and Hall (1992) propose an alternative which is an approximation to least-squares estimation in (4.1), but suffers from greater sampling variability in the terms of $\mathbf{L}$, as shown in Figure 2. That sampling variability translates into less accurate curvilinear relationships between T^* and L^*.

4.1 Tail-specific regression approximations

We considered a variety of local and global regression methods for tail-specific approximations. The local methods did not work well, suffering from excessive sampling variability, because they are based on only a small fraction of all bootstrap samples (e.g. the $2\alpha B_0$ replications with the largest values of T^*). Global regression methods suffer from other problems, but those can be fixed.

Our global regression principle is to fit a nonlinear surface using all the bootstrap samples, and use the gradient of that surface at an appropriate point in the tail. However, fitting a quadratic relationship between T^* and $\mathbf{P}^*$ requires estimating $n-1$ first derivatives ($\mathbf{P}$ is of rank $n-1$) and $n(n-1)/2$ second derivatives, which may be too many coefficients to estimate with a modest bootstrap sample size. Our solution is to fit a restricted quadratic surface, estimating only the linear combination of second derivatives which affects the gradient at the tail evaluation point. Details of the restricted quadratic fitting are available from the author. It

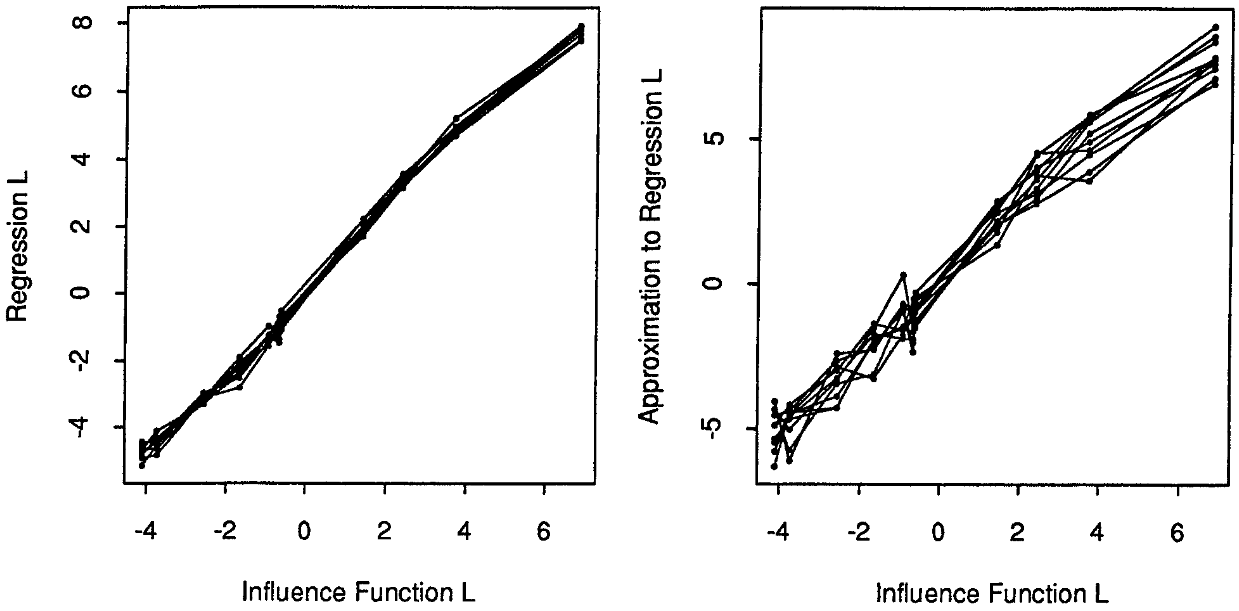


Figure 2: Central Regression $\mathbf{L}$ for the t-statistic and an approximation.

10 vectors $\mathbf{L}$, each estimated from 500 bootstrap samples.

works moderately well, but suffers from sampling variability. Estimating the nonlinear transformation ψ and fitting $\hat{\psi}(T^*)$ (instead of T^*) as a restricted quadratic function of $\mathbf{P}^*$ reduces the sampling variability considerably, with improvement comparable to using the full regression rather than the approximation in Figure 2.

5 Transformations of T^*

We have nearly concluded our discussion of linear approximations, and are about ready to turn to applications. But first we turn to a topic which applies to both the approximations and applications, that of estimating the nonlinear transformation $\psi(T^*)$ in (2.1). The butcher knife and regression approximations (and to a lesser extent the jackknife approximations) are not invariant under nonlinear transformations, so we can improve those estimates by replacing T with an estimate $\hat{\psi}(T)$. And the concomitants technique below is not equivariant under nonlinear transformations, and can be improved considerably using an estimate of ψ. A third use for a transformation is for improving the bootstrap-t confidence interval; see Tibshirani (1988).

We propose two ways to estimate ψ, one deterministic, the other based on bootstrap observations. The former requires extra evaluations of T^*, while the latter gives estimates which are subject to sampling variability.

The deterministic procedure is to estimate ψ by interpolating $L = \mathbf{P} \cdot \mathbf{L}$ (as y) against $T(\mathbf{P})$ (as x), for points $\mathbf{P}$ determined using exponential tilting (3.4); every distinct value of the tilting parameter τ results in one training point for the interpolation.

The nondeterministic procedure is to estimate ψ using a scatterplot smooth or nonlinear regression of L^* (as y) against T^*, for B_0 bootstrap samples. This is motivated by the ACE algorithm of Breiman and Friedman (1985), and is nearly invariant under ψ, subject to limitations of the smoothing method. The estimate should then be rescaled so that $\hat{\psi}'(T(\mathbf{P}_0)) = 1$.

Both the deterministic and nondeterministic transformations nicely remove the curvilinearity observed in Figure 1, and give much better regression linear approximations, with sampling variability in the elements of $\mathbf{L}$ smaller by factors of approximately four and six for the central and tail-specific approximations, respectively.

6 Variance Reduction

We have three purposes in this section — to justify the effort put into accurate linear approximations by showing how sensitive variance reduction techniques are to that accuracy, to indicate when tail-specific approximations will give a significant improvement, and to propose a new variance reduction technique that uses linear approximations.

Figure 3 shows the relative efficiency for the three variance reduction techniques as a function of the correlation between T^* and L^*. Note how quickly the concomitants and stratified sampling procedures lose efficiency as the correlation drops. This figure indicates why more accurate linear approximations are worth pursuing.

The "efficiency" in Figure 3 and later is the efficiency of a technique relative to simple bootstrap sampling, for estimating tail probabilities corresponding to the 0.025 and 0.975 percentiles of the distribution of T^*. See Hall (1991) and Johns (1988) for asymptotic results in special cases that indicate that this efficiency is asymptotically equivalent to the efficiency for estimating the percentiles themselves.

We assume in Figure 3 that the relationship between T^* and L^* is linear, that L^* is a central approximation, and that the distribution of L^* is normal, but do not assume that the joint distribution is bivariate normal — it was very clearly not in Figure 1. Instead, assuming that T^* can be written as a smooth function of sample means, we find that the joint distribution is of the form

$$\psi(T^*) \doteq L^* + (L^* - L_0)n^{-1/2}\sum_{d=1}^{D} a_d Z_d + n^{-1/2}\sum_{d=1}^{D} b_d Z_d^2 \qquad (6.1)$$

for some ψ, where $(L^* - L_0) = O_P(n^{-1/2})$, the Z_d are independent standard normal random variables, L_0 depends on the linear approximation, and D is a small integer that depends on the statistic, not on n. We omit the formal statement and proof of this theorem in this version of this article. It turns out that tail-specific linear approximations work best when the a's are large compared to the b's, and it is clear from (6.1) that the conditional variance of $\psi(T^*)$ given L^* is smallest in the center when any of the a's are nonzero. So a scatterplot of T^* against L^* can be used to diagnose if a tail-specific approximation may be useful, without actually computing it; a small conditional variance in the center, as in Figure 1, indicates that a tail-specific approximation can help.

The two panels in Figure 3 correspond to the "heteroskedastic normal" case where $b_d = 0$ for $d = 1, \dots D$, and the "single-χ^2" case where $b_1 \neq 0$ and the other a's and b's are zero. In the heteroskedastic normal case the conditional distribution of $\psi(T^*)$ given L^* is normal with standard deviation proportional to $|L^* - L_0|$; we see similar behavior in Figure 1. The heteroskedastic normal case is interesting because it is offers the greatest potential for tail-specific linear approximations, and because without tail-specific approximations it is a hard case for the concomitants and stratified sampling procedures.

The single-χ^2 case is also interesting because it has the heaviest tails of all conditional distributions of $\psi(T^*)$ given L^*, for fixed ρ and distributions in the family (6.1). That those heavy tails are a problem for importance sampling is apparent in the right panel of Figure 2. Unfortunately, tail-specific approximations offer little help here, but the damage can be mitigated by using more conservative importance sampling.

6.1 Concomitants of order statistics

For simplicity of notation, sort the bootstrap samples by the values of L_b^*. Then the concomitants estimate of the bootstrap distribution is $\hat{G}(a) = (1/B)\sum_{b=1}^{B} I(T_b^\dagger \leq a)$. where

$$T_b^\dagger = T_b^* + L_b^\dagger - L_b^* \qquad (6.2)$$

where $L_b^\dagger$ is an estimate of the $(b-0.5)/B$ quantile of the distribution of L^*. Efron (1990) lets $L_b^\dagger$ be the b'th normal score $\Phi^{-1}((b-0.5)/B)$, iteratively transformed using cubic Cornish-Fisher transformations so that the first four sample moments match the theoretical moments of L^*, but suggests that letting $L_b^\dagger$ be the saddlepoint estimate of $L^{*((b-0.5)/B)}$ would be more accurate.

In the case that L^* and T^* are jointly continuous we find the asymptotic variance of the concomitants estimate to be

$$B\mathrm{Var}(\hat{G}(a)) \doteq \int H(a-l|l)(1-H(a-l|l))f(l)dl + 2\int_{l_1<l_2} H'(a-l_1|l_1)H'(a-l_2|l_2) F(l_1)(1-F(l_2))dl_1dl_2 \qquad (6.3)$$

where F and f are the distribution and density functions of L^*, $H(a|l) = P(T^* \leq a|L^* = l)$, and $H'(a|l) = \frac{\partial}{\partial l}H(a|l)$. Note, in Figure 3, how strongly the efficiency depends on the correlation between L^* and T^*.

The concomitants procedure is not invariant under nonlinear transformations T^*. If L^* is a good approximation to some transformation $\psi(T^*)$ rather than to T^* the double integral in (6.3) can be substantial. Efron (1990) replaced the right side of (6.2) with $T_b^* + \hat{\psi}^{-1}(L_b^\dagger) - \hat{\psi}^{-1}(L_b^*)$ for the t-statistic, with ψ^{-1} estimated using cubic regression of T^* against L^*. We suggest replacing the right side of (6.2) with $\hat{\psi}^{-1}(\psi(T_b^*) + L_b^\dagger - L_b^*)$ instead, which is invariant under transformations of T^* (up to limitations of the procedure used to estimate ψ).

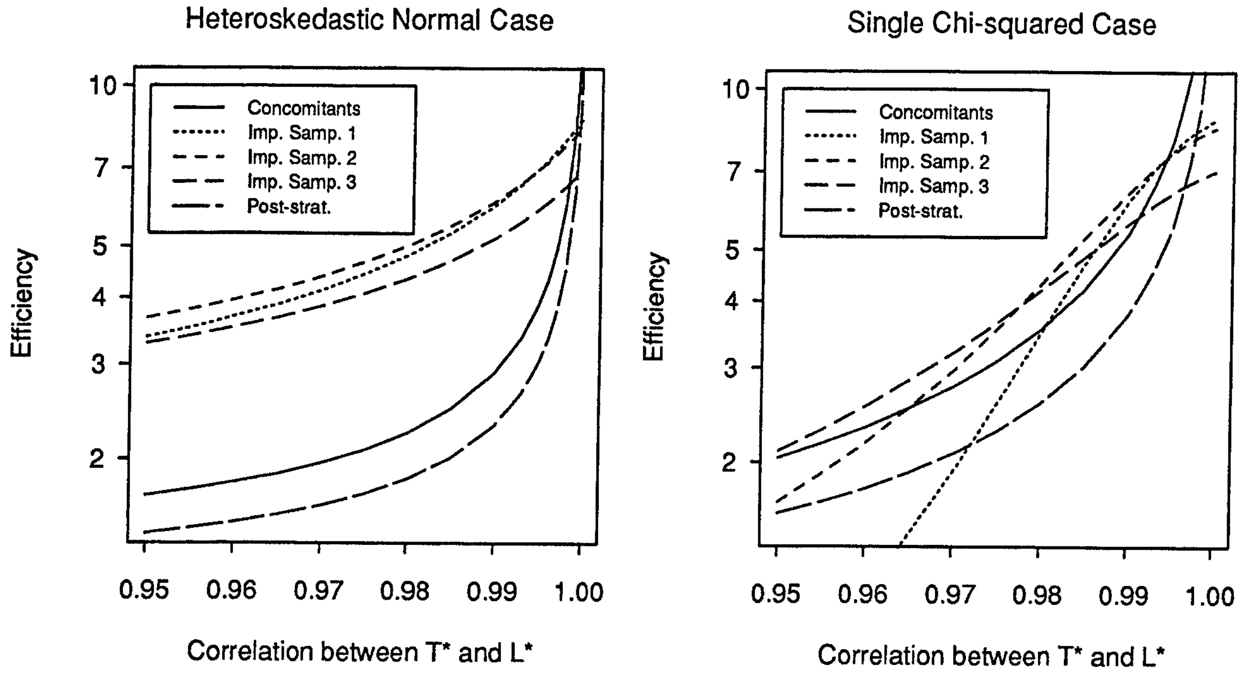


Figure 3: Efficiency for Tail Estimation as a function of the correlation between L^* and T^*.

We show three variations of the concomitants procedure in Table 1. The first variation uses (6.2), the second uses Efron's procedure with cubic regression, and the third uses the invariant procedure with ψ estimated using the deterministic procedure in section 5. In all cases we use the saddlepoint estimate of the inverse cumulative distribution function of L^* (Hesterberg 1994) to determine $L_b^\dagger$. The transformations do result in higher efficiency, with the invariant procedure performing best, but the biggest improvement is obtained by using tail-specific linear approximations rather than central linear approximations. The efficiencies using tail-specific approximations are three to four times higher than those obtained using central approximations.

6.2 Importance sampling

Importance sampling uses bootstrap samples of size n generated from a distribution g rather than by simple random sampling f, and places weight $W_b = (1/B)f(\mathcal{X}_b^*)/g(\mathcal{X}_b^*)$ on T_b^* to counteract the sampling bias, resulting in the distribution function estimate

$$\hat{G}(a) = \begin{cases} \sum_{b=1}^{B} W_b I(T_b^* \le a) & \text{left tail} \\ 1 - \sum_{b=1}^{B} W_b I(T_b^* > a) & \text{right tail} \end{cases} \quad (6.4)$$

The multi-part definition is necessary because the weights do not add to 1, and $\hat{G}$ is inaccurate in the center; see Hesterberg (1988, 1991) for further discussion and methods suitable for problems other than tail estimation.

We consider sampling distributions g of the form

$$g_\Lambda(\mathcal{X}^*) = \lambda_0 f(\mathcal{X}^*) + \lambda_1 g_1(\mathcal{X}^*) + \lambda_2 g_2(\mathcal{X}^*)$$

where the λ's are nonnegative mixing proportions that add to 1 and g_k indicates sampling with unequal probabilities $P\{X_i = x_j\} = c_k \exp(\tau_k L_{j,k})$ for $k = 1, 2$ respectively, where the c_k are normalizing constants. This is a combination of exponential tilting (Johns 1988, Davison 1988) with defensive mixture distributions (Hesterberg 1988, 1991), and we stratify the mixing proportions by drawing exactly $B_0 = \lambda_0 B$ bootstrap samples using simple random sampling and $B\lambda_k$ samples using g_k. The weight

$$W_b = B^{-1} \frac{1}{\lambda_0 + \lambda_1 c_1^n \exp(n\tau_1 L_2^*) + \lambda_2 c_2^n \exp(n\tau_2 L_2^*)}$$

is independent of which distribution was used to generate sample b. The variance of the distribution estimate is

$$\text{Var}(\hat{G}(a)) = \frac{1}{B}\sum_{k=0}^{2} \lambda_k \text{Var}_{g_k}(WI(T^* \text{ exceeds } a)),$$

where "exceeds" means "$\le$" and "$>$" for right and left tails, respectively.

	Left Tail—($\alpha = 0.025$)		Right Tail—($\alpha = 0.975$)	
	Central	Tail-specific	Central	Tail-specific
Concomitants 1	1.5	5.6	2.5	8.6
Concomitants 2	2.0	6.3	2.4	9.0
Concomitants 3	2.1	8.0	2.5	10.8
Importance Sampling 1			8.0	16.6
Importance Sampling 2	5.3	12.5	4.9	8.9
Importance Sampling 3	4.4	9.8	4.3	7.2
Post-Stratification	1.5	4.7	1.8	5.6

Table 1: Efficiency using Central and Tail-Specific Linear Approximations

Estimated efficiency for estimating tail probabilities, for $B = 200$, for the t-statistic, using the central and tail-specific influence function linear approximations. The importance sampling distributions parallel those in Figure 3, but with values of τ chosen so that $\mathbf{L} \cdot \mathbf{P}_\alpha = L^{*(\alpha)}$, using saddlepoint quantiles (Hesterberg 1994). The estimates are based on 2000 bootstrap experiments. Standard errors are less than 4% of the estimates, except for the tail-specific post-stratification estimates (less than 7%).

The first sampling distribution in Figure 3 uses simple exponential tilting ($\lambda_2 = 1$) rather than a mixture, with $\tau = 2.18/\mathrm{Var}(L^*)$, which Johns (1988) finds to be optimal for the 97.5 %-tile when $T^* = L^*$ and L^* is normal. This is a very anti-conservative sampling distribution, as f/g is practically unbounded, and is not robust to imperfect linear approximations, particularly in the heavy-tailed single-χ^2 case.

The second distribution uses $B_1 = B_2 = B/2$ bootstrap samples computed using exponential tilting for the two tails, with $-\tau_1 = \tau_2 = z_\alpha/\mathrm{Var}(L^*)$. The third uses $B_0 = B/5$ simple random bootstrap samples, and $B_1 = B_2 = 0.4B$ samples using the same τ_1 and τ_2. These distributions are more conservative (with W_b bounded above by approximately $6.8/B$ and $3.2/B$, respectively), and do slightly worse when $\rho = 1$, but do much better for smaller ρ. A description of optimal choice of the τ's and λ's is beyond the scope of this article, but we will note that the second and third distributions are more conservative (larger λ_0 and smaller τ's) for $\rho > 0.95$ than is optimal if (4.5) holds exactly, but do offer insurance against the effect of cubic and higher deviations from linearity, which result in heavier tails than a χ^2 random variable.

Two practical considerations argue in favor of the second or third importance sampling distributions. Many bootstrap methods require estimates of quantiles from both tails; a mixture that incorporates g_{left} and g_{right} is a robust and more efficient alternative to performing separate simulations for each tail. Second, if importance sampling is combined with linear regression methods for determining $\mathbf{L}$, the B_0 bootstrap samples from f may be used as the training set for the regression.

The tail-specific linear approximations result in substantially better efficiency in Table 1, roughly by a factor of two.

6.3 Post-stratification

We propose $I(L^* \leq L^{*(\alpha)})$ as a variable for post-stratification, and let

$$\hat{G}(a) = \sum_{b=1}^{B} W_b I(T_b^* \leq a)$$

be the empirical distribution formed by placing weight

$$W_b = \begin{cases} \frac{P\{L^* \leq L^{*(\alpha)}\}}{\#(L^* \leq L^{*(\alpha)})} & \text{if } L_b^* \leq L^{*(\alpha)} \\ \frac{P\{L^* > L^{*(\alpha)}\}}{\#(L^* > L^{*(\alpha)})} & \text{if } L_b^* > L^{*(\alpha)} \end{cases} \tag{6.5}$$

on T_b^*. The estimate is unbiased with variance

$$\begin{aligned} \mathrm{Var}(\hat{G}(a)) &= \frac{h(P\{T^* < a | L^* \leq L^{*(\alpha)}\})}{\#(L^* \leq L^{*(\alpha)})} \\ &+ \frac{h(P\{T^* < a | L^* > L^{*(\alpha)}\})}{(\#L^* > L^{*(\alpha)})} \end{aligned}$$

conditional on $\#(L^* \leq L^{*(\alpha)})$, where $h(p) = p(1-p)$ is the variance of a Bernoulli random variable with mean p, and is asymptotically normal with asymptotic standardized variance

$$\begin{aligned} &\alpha^{-1} h(P\{T^* < a | L^* \leq L^{*(\alpha)}\}) \\ &+ (1-\alpha)^{-1} h(P\{T^* < a | L^* > L^{*(\alpha)}\}) \end{aligned}$$

as $B \to \infty$. At $a = T^{*(\alpha)}$ this reduces to $2p_{1,2} - p_{1,2}^2/(\alpha(1-\alpha))$ where $p_{1,2} = P\{L^* \leq L^{*(\alpha)}, T^* > T^{*(\alpha)}\}$.

Under fairly general conditions, including the smooth functions of means model considered in Do & Hall (1992), the errors in linear approximations are such that $p_{1,2} = O(n^{-1/2})$. Thus at $a = T^{*(\alpha)}$, this post-stratification estimator shares the same factor of $n^{-1/2}$ in the variance as the concomitants procedure. The process may be repeated for for every level α for which an estimate is desired.

Post-stratification requires an estimate of $L^{*(\alpha)}$, which we obtain using the saddlepoint quantile estimate of Hesterberg (1994), based on the saddlepoint formula of Lugannani and Rice (Daniels 1987). Davison and Hinkley (1988) use the saddlepoint for linear bootstrap problems; post-stratification uses a linear approximation for nonlinear problems.

Post-stratification does not do as well as either concomitants or importance sampling in Table 1. On the other hand, it is substantially simpler than those procedures. It uses simple random sampling, does not require an estimate of ψ, and requires only one saddlepoint estimate (of $L^{*(\alpha)}$) for each quantile desired.

7 Conclusion

The central and tail-specific influence function and positive jackknife linear approximations work well in the t-statistic example and in other problems we investigated where the bootstrap statistic T can be written as a smooth function of weights, including the bivariate correlation coefficient and sample variance, although in the latter cases the gains from tail-specific approximations are less than with the t-statistic. The choice between the influence function and positive jackknife reduces to a question of implementation, whether the analytical calculations required by the influence function or the numerical calculations required by the jackknife are easier.

The butcher knife worked nearly as well in the smooth function problems, and also worked for the sample median and 25% trimmed mean. However, the butcher knife is a radical proposal. Where each numerical calculation for the (central) positive jackknife is equivalent to repeating a single observation twice, the butcher knife corresponds to repeating the observation $n^{1/2}$ times; this would be too many for statistics which are sensitive to the number of times observations are repeated.

The central regression approximation calculated from T^* worked reasonably well, but the tail-specific version suffered from excessive sampling variability. Both versions were significantly improved by replacing T^* with $\hat{\psi}(T^*)$.

If estimates for multiple values of α in each tail are needed it should suffice to use a single tail-specific linear approximation for all. Particularly for importance sampling it would be impractical to use multiple approximations and multiple sampling distributions for each tail.

Simulation Details

Simulations are run in S (version S-PLUS 3.0) (Becker et. al., 1988; Statistical Sciences, 1991) and C, using the Super Duper random number generator of Marsaglia, using common random numbers with the original observations sorted. Antithetic variates and balancing are not used.

References

Becker, R.A., Chambers, J.M., and Wilks, A.R. (1988), *The New S Language*, Pacific Grove, CA: Wadsworth and Brooks/Cole.

Beran, R. J. (1987). "Prepivoting to reduce level error of confidence sets," *Biometrika* 74, 457-468.

Booth, J.G. & Hall, P. 1992. "Monte Carlo Approximation and the Iterated Bootstrap," Technical Report CMA-SR30-92, The Australian National University.

Breiman, L. and Friedman, J.H. (1985) "Estimating Optimal Transformations for Multiple Regression and Correlation" (with discussion), *Journal of the American Statistical Association* **80**, 580-619.

Daniels, H.E. (1987). "Tail Probability Approximations," International Statistical Review **55**, 1, 37-48.

Davison A.C. (1988) Discussion of paper by D.V. Hinkley, *Journal of the Royal Statistical Society, Series B* **50** 356-57.

Davison A.C. and Hinkley D.V. (1988) "Saddlepoint Approximations in Resampling Methods," *Biometrika* **75** 417-31.

Davison A.C, Hinkley D.V., & Schechtman E (1986). "Efficient Bootstrap Simulation," *Biometrika* **74** 555-66.

Do, K. & Hall, P. (1991). "On importance resampling for the Bootstrap," *Biometrika* **78**, 1, 161-167.

Do, K. & Hall, P. (1992). "Distribution Estimation using Concomitants of Order Statistics, with Application to Monte Carlo Simulation for the Bootstrap," *Journal of the Royal Statistical Society, Series B* **54**, 2, 595-607.

Efron, B. (1982). *The Jackknife, the Bootstrap and Other Resampling Plans.* Society for Industrial and Applied Mathematics, Philadelphia.

Efron, B. (1987). "Better Bootstrap Confidence Intervals (with discussion)," *Journal of the American Statistical Society* **82**, 397, 171-200

Efron, B. (1990). "More Efficient Bootstrap Computations," *Journal of the American Statistical Society* **85** 79-89.

Efron, B., and Tibshirani, R (1993). *An Introduction to the Bootstrap.* Chapman & Hall, New York.

Graham, R.L., Hinkley D.V., John P.W.M., & Shi, S. (1990) "Balanced Design of Bootstrap Simulations," *Journal of the Royal Statistical Society, Series B* **52** 185-202.

Hall, P. (1986), "On the Number of Bootstrap Simulations Required to Construct a Confidence Interval," *Annals of Statistics* **14** (4) 1453-1462.

Hall, P. (1991), "Bahadur Representations for Uniform Resampling and Importance Resampling, with Applications to Asymptotic Relative Efficiency," *Annals of S-tatistics* **19** (2) 1062-1072.

Hesterberg, T. C. (1987), "Importance Sampling in Multivariate Problems," *Proceedings of the Statistical Computing Section, American Statistical Association 1987 Meeting*, 412-417.

Hesterberg, T. C. (1988), "Advances in Importance Sampling," Ph.D. dissertation, Statistics Department, Stanford University.

Hesterberg, T.C. (1991) "Weighted Average Importance Sampling and Defensive Mixture Distributions," Technical Report No. 148, Division of Biostatistics, Stanford University.

Hesterberg, T.C. (1994) "Saddlepoint Quantiles and Distribution Curves, with Bootstrap Applications," to appear in *Computational Statistics.*

Johns M.V. (1988). "Importance Sampling for Bootstrap Confidence Intervals," *Journal of the American S-tatistical Association* **83**, 403, 709-714.

Lugannani, R., and Rice, S. (1980). "Saddle Point Approximation for the Distribution of the Sum of Independent Random Variables," *Advances in Applied Probability* **12**, 475-490.

Statistical Sciences, Inc. (1991), *S-PLUS Reference Manual, Version 3.0,* Seattle: Statistical Sciences, Inc., 1991.

Tibshirani, R. (1988). "Variance Stabilization and the Bootstrap," *Biometrika* **75**, 3, 433-444.

Efficient simulation from the Random walk Metropolis algorithm

G.O. Roberts
Statistical Laboratory
University of Cambridge
Cambridge CB2 1SB
UK

Abstract

Perhaps the most popular Markov chain Monte Carlo method from the class of Hastings-Metropolis algorithms, is the symmetric random walk Metropolis algorithm. This paper will discuss some of its theoretical properties. Conditions ensuring geometric convergence of the algorithm will be given, in terms of smootheness and exponential decay conditions on the target distribution, and an example where geometric ergodicity does not happen is discussed. Finally, recent results on optimal scaling of proposal kernels as a function of dimension of the target distribution will be given, and the results related to overall acceptance rates of the algorithm.

1 Introduction

It is now well understood that the convergence properties of the Gibbs sampler (see for example Gelfand and Smith, 1990), are closely linked to the correlation structure of functionals of coordinate directions (see for example Amit, 1991, Hills and Smith 1992). Unfortunately, the Metropolis algorithm (Metropolis et. al., 1953) is considerably less well understood. In particular, there is no obvious connection between convergence rates of Metropolis algorithms, and the statistical propoerties of target densities. Although known to work well very often in practice, Metropolis algorithms are not automatic preocedures - a proposal density needs to be chosen apriori - and the choice of proposal can often be critical to the efficiency of the algorithm.

Very little progress has been made on the problem of choosing a proposal, even for the simplest algorithm, the random walk Metropolis algorithm. A number of authors have suggested scaling the proposal variance in proportion to the variance of the target density. In practice, it is impossible to *apriori* obtain a reliable estimate of the target variance, so that Tierney (1991) and Muller (1993) suggest monitoring the proportion of accepted Metropolis jumps. This is an appealing approach since the acceptance rate, p_{jump}, is extremely easy to monitor. Muller observes that an acceptance rate of around 0.5 often works well, but can any theoretical justification be given for using such rules?

This paper will discuss two sets of results. First of all, we consider the problem of determining when the random walk Metropolis algorithm is geometrically ergodic. It turns out that geometric ergodicity is related to the tail behaviour of the target density, and to a curvature condition on the contours of the target density, but that the form of the proposal density is (essentially) unimportant. Section 2 discusses these results; further details and proofs appear in Roberts and Tweedie (1994a).

The second set of results consider a diffusion approximation for high dimensional Metropolis algorithms with spherically symmetric proposal densities. The limiting diffusion process has a speed measure which we can interpret as the asymptotic efficiency of the algorithm. This speed measure depends only on the scaling of the proposal density variance, and this in turn can be related to p_{jump}. Thus, efficiency can be related directly to p_{jump}, and it turns out that the optimal value for p_{jump} should be somewhere around 0.25. Perhaps more usefully in practice, an acceptance rate of between 0.15 and 0.4 gives at least 85% of maximal possible efficiency. These results are covered in Section 3; further details including proofs and practical implications of these results appear in Roberts Gelman and Gilks (1994) and Gelman Roberts and Gilks (1994).

2 Geometric convergence of the Random walk Metropolis algorithm

We say that a Markov chain X with state space contained in $\mathbf{R}^d$ is geometrically ergodic (to π) in *total vari-*

ation norm, if π is a probability measure and

$$\int_{\mathbf{y}\in\mathbf{R}^d} |P^t(\mathbf{x},\mathbf{y}) - \pi(\mathbf{y})| d\mathbf{y} \leq V(\mathbf{x})\rho^t$$

$\forall \mathbf{x} \in \text{supp}\pi$. Here, P^t denotes the t-step transition kernel for X, V is a real-valued function, and $\rho < 1$.

We argue here that geometric convergence is a minimal, but important requirement that should be satisfied by a Markov chain Monte Carlo algorithm. Ideally, we would like quantitative bounds on V and ρ. However here we content ourselves with qualitative results because

(1) Quantitative bounds for relatively complex problems are extremely difficult to obtain (although see Rosenthal, 1994).

(2) Non geometric algorithms have heavy tailed excursion, so have a tendency to *get stuck*. This can also make the choice of starting value highly critical.

(3) Geometric convergence results at least allow the existence of central limit theorems (see for example Roberts and Tweedie, 1994a), allowing some reassurance for the use of ergodic estimates of Markov Chain Monte Carlo output.

(4) Surprisingly perhaps, many of the algorithms commonly used (including in some cases, the random walk Metropolis algorithm) fail to be geometrically ergodic.

The following result can be used to demonstrate either geometric or non-geometric convergence. We do not state it in its most general form, although we will need the following definitions. We say that a set C is *small* if there exists $\epsilon > 0$, $t \in \mathbf{N}$, and a probability measure $\nu(\cdot)$ such that

$$\sup_{\mathbf{x}\in C} P^t(\mathbf{x}, A) \geq \epsilon\nu(A)$$

for all sets A. In our context compact sets are nearly always small, although it is frequently possible for unbounded sets to be small also.

Let $\tau_C = \inf\{t \geq 1;\ \mathbf{X}_t \in C\}$.

Theorem 1 *(Meyn and Tweedie, 1993) The following three statements are all equivalent.*

(1) X is geometrically ergodic.

(2) (Foster drift condition) There exists a small set C, a function $V \geq 1$. $\lambda < 1$, and $b \geq 0$ such that

$$\mathbf{E}_{\mathbf{x}}[V(\mathbf{X}_1)] \leq \lambda V(\mathbf{x}) + bI[\mathbf{x} \in C].$$

(3) There exists a small set C and a constant $\kappa > 1$ such that

$$\sup_{\mathbf{x}\in C} \mathbf{E}_{\mathbf{x}}[\kappa^{\tau_C}] < \infty.$$

The second equivalence is most useful for demonstrating geometric convergence whereas the third is particularly useful for establishing that an algorithm is not geometrically ergodic. The following result is a simple consequence of (3).

Theorem 2 *(Roberts and Tweedie, 1994a) A necessary condition for geometric convergence of a Markov chain with stationary distribution π, not concentrated at a single point is that*

$$ess\ sup P(\mathbf{x}, \{\mathbf{x}\}) < 1.$$

Here the essential supremum is taken with respect to the stationary distribution π.

We will now describe results for the random walk Metropolis algorithm which can be derived from Theorems 1 and 2. For simplicity (although this is not necessary in most of the results that we give), we shall assume that the random walk step is a spherically symmetric continuous distribution, and that the *target density* π from which we wish to sample, is a d-dimensional Lebesgue density on $\mathbf{R}^d$. Let q denote the proposal kernel, and suppose α be the acceptance probability of any particular move. Therefore, the algorithm proceeds iteratively as follows. Given $\mathbf{X}^{t-1}$, choose $\mathbf{Y}$ according to the density $q(|\mathbf{Y} - \mathbf{X}^{t-1}|)$. Accept $\mathbf{Y}$ and set $\mathbf{X}_t = \mathbf{Y}$ with probability

$$\min\{1, \frac{\pi(\mathbf{Y})}{\pi(\mathbf{X}^{t-1}}\}.$$

Otherwise set $\mathbf{X}^t = \mathbf{X}^{t-1}$.

Roberts and Tweedie (1994a) gives very general conditions for geometric ergodicity of Hastings-Metropolis algorithms, and in particular the random walk Metropolis algorithm. We content ourselves with a brief summary of the main ideas. Therefore, regularity and smootheness conditions are omitted, as well as the most general statement of the result.

Define

$$C_\epsilon = \{\mathbf{x};\ \pi(\mathbf{x}) = \epsilon\}$$

to be the contour of ϵ for (typically) small ϵ. Now define $\kappa(\epsilon)$ to be the supremum of the *Ricci* curvature over all points on C_ϵ. The Ricci curvature is the multidimensional analogue of curvature, and can be described in terms of the curvature of the largest hypersphere that can be inscribed (locally at least) within the interior of the contour manifold.

Essentially, the random walk Metropolis algorithm is geometrically ergodic when the following two conditions are satisfied:

(A)

$$\lim_{\epsilon \to 0} C_\epsilon = 0.$$

(B) There exists constants A, $B > 0$ such that

$$\pi(\mathbf{x}) \leq Ae^{-B|\mathbf{x}|}.$$

Under these conditions, the test function $V(\mathbf{x}) = \pi(\mathbf{x})^{-1/2}$ can be used in the second equivalence of Theorem 2.

The exponential decay condition (B) is close to being necessary for geometric convergence. Intuitively, for target dentities with tails heavier than exponential. there is too much mass in the tails for a random walk dynamic to be able to see quickly enough. In one dimension (A) is not relevant, and Mengerson and Tweedie (1993) essentially demonstrate necessity and and sufficiency of (B) for geometric convergence of the algorithm.

Condition (A) is far from being a necessary condition, although some kind of restriction on $\kappa(\epsilon)$ for small ϵ is necessary as the following example demonstrates.

Example 1 Suppose

$$\pi(\mathbf{x}) \propto \exp\{-x^2 - x^2y^2 - y^2\},$$

then it is not hard to show that $\kappa(\epsilon) \to \infty$ as $\epsilon \to 0$. Therefore the density has long *ridges* along the coordinate axes. The random walk Metropolis algorithm can be shown to be not geometrically ergodic by Theorem 2. (See Roberts and Tweedie, 1994a for further details).

However large classes of densities can be shown to satisfy (A) and (B).

Example 2 Suppose $\pi(\mathbf{x})$ is positive everywhere, and satisfies

$$\pi(\mathbf{x}) = t(\mathbf{x}) \exp\{-r(\mathbf{x})\}$$

where t and r are polynomials and r satisfies the following "positive definiteness" property. Suppose r is of degree m, then if r_m is the polynomial consisting of all r's m-th order terms, $r_m(\mathbf{x}) \to \infty$ as $\mathbf{x} \to \infty$. Then π satisfies (A) and (B), and so the random walk Metropolis algorithm is geometrically ergodic.

π is example 1 fails to satisfy the positive definiteness condition since $r_m = x^2y^2$.

3 Efficiency and scaling of proposals

To fix ideas in this section, we shall assume that the proposal distribution is normal, so that

$$q(\mathbf{x}, \mathbf{y}) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left\{ - \frac{(\mathbf{y} - \mathbf{x})^T(\mathbf{y} - \mathbf{x})}{2\sigma^2} \right\}.$$

The question we address here is: how should we choose σ to make the algorithm as efficient as possible?

Unfortunately this question is illposed - there is no unique measure of efficiency for such an algorithm. A discussion of different measures of efficiency appears in Gelman Roberts and Gilks (1994). Instead, we use an asymptotic argument as d gets large. Consider first the case where the d-dimensional target density π_d has the product form:

$$\pi_d(\mathbf{x}) = \prod_1^d f(x_i). \qquad (1)$$

For the d-dimensional problem, choose $\sigma = \phi/\sqrt{d}$. It turns out that this is the right way of scaling the variance. Now define

$$Y_t^d = X_1^{[dt]}$$

to be a *speeded up* version of the first component of $\mathbf{X}$. Now Y^d is making smaller and smaller jumps, more and more often, so that if a sensible limit process exists as $d \to \infty$, then we would expect it to be a continous process. Although Y^d is not Markov for any d, the limiting process turns out to be a Langevin diffusion, and is hence Markov. The limiting process satisfies the stochastic differential equation

$$dY_t = \frac{f'(Y_t)h(\phi)}{2f(Y_t)} dt + h(\phi)^{1/2} dB_t, \qquad (2)$$

where

$$h(\phi) = 2\phi^2 \Phi\left(\frac{-\phi F^{1/2}}{2}\right), \qquad (3)$$

and

$$F = \int_{-\infty}^{\infty} \frac{(f'(x))^2}{f(x)} dx \qquad (4)$$

is a Fisher's information measure for f ($F = 1$ for standard normal f). The limiting value of p_{jump} for this sequence of problems is $h(\phi)/\phi^2$.

The speed of the diffusion $h(\phi)$ is maximized by the choice

$$\phi = \tilde{\phi} = 2.38/F^{1/2}.$$

Therefore the asymptotically optimal jumping kernel has variance-covariance matrix $(\tilde{\phi}^2/d)I_d$, with jumping probability approximately 0.234.

This is the simplest of such results. Many more general forms of target density are subject to similar results. In particular, it is not necessary for π to have the product form of (1). Moreover, the asymptotically optimal acceptance rate of 0.234 remains robust to many generalizations. These extensions and the proof of the above result appear in Roberts Gelman and Gilks (1994).

4 Summary and Conclusions

The random walk Metropolis algorithm is often thought of as a default option: it is easy to implement, and it requires and uses no information about the structure of the target density being sampled. As such, for specific problems, there is frequently a more efficient algorithm available. However in return, the random walk Metropolis algorithm has relatively robust theoretical properties, as discussed in Section 2. In contrast, more 'tailor-made' algorithms such as those derived from Langevin diffusion approximations can have highly undesirable properties (see Roberts and Tweedie, 1994b).

The efficiency results of Section 3 suggest that the algorithm should perform best with overall acceptance rates in the range [0.15, 0.4]. However a number of words of caution are in order.

There are many target densities for which the random walk Metropolis algorithm is inappropriate, for instance highly multi-modal or heavy tailed distributions. The results of Section 3 only provide an efficiency measure *relative* to other random walk Metropolis algorithms. There is no guarantee that there exists a proposal scaling that gives an *absolutely* efficient algorithm.

The result is asymptotic, and although it is supported by simulation studies for relatively well-behaved unimodal densities (see Gelman, Roberts and Gilks, 1994), it's performance on low-dimensional multimodal problems is unlikely to be satisfactory.

In practice, one might try to "fine-tune" the algorithm as the simulation proceeds in order to obtain actual acceptance rates within the range suggested. Care has to be taken with such a procedure since the stationarity of the target density can be compromised by such a non-Markov procedure (for example see Gelfand and Sahu, 1993). Moreover, observed acceptance rates may be misleading in an inefficient algorithm. Therefore monitoring acceptance rates should never be used as a diagnostic for efficiency.

Acknowledgements

I am grateful to Richard Tweedie, Wally Gilks, and Andrew Gelman for fruitful discussions and collaboration on work reported in this paper.

References

Amit Y (1991). On rates of convergence of stochastic relaxation for Gaussian and non-Gaussian distributions. *J Mult Anal* **38**, 82–99.

Gelfand A. E., and Sahu S. (1993). On Markov chain Monte Carlo acceleration. To appear, *J. Comp. Graph. Statist.*

Gelfand, A.E. and Smith, A.F.M. (1990). Sampling based approaches to calculating marginal densities. *J. Amer. Stat. Assoc.* **85**, 398–409.

Gelman, A. Roberts, G.O. and Gilks, W.R. (1994), Efficient Metropolis jumping rules, submitted to Bayesian Statistics 5, Valencia, Spain.

Hills, S. E., and Smith, A. F. M. (1992). Parameterization issues in Bayesian inference (with discussion). In *Bayesian Statistics 4*, ed. J. Bernardo, Oxford University Press, 227–246.

Mengersen K.L. and Tweedie R.L. (1993). Rates of convergence of the Hastings and Metropolis algorithms. Submitted for publication.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092.

Meyn S.P. and Tweedie R.L. (1993). *Markov Chains and Stochastic Stability.* Springer-Verlag, London, 1993.

Muller, P. (1993). A generic approach to posterior integration and Gibbs sampling. *J. Amer. Stat. Assoc.*, to appear.

Roberts, G. O., Gelman, A., and Gilks, W.R. (1994). Weak convergence and optimal scaling of random walk Metropolis algorithms. Research report.

Roberts G.O and Tweedie R.L. (1994a). Geometric Convergence and Central Limit Theorems for Multidimensional Hastings and Metropolis Algorithms. Cambridge University Statistical Laboratory reseach report 94-??

Roberts G.O and Tweedie R.L. (1994b). MCMC using simulation from Langevin diffusions and their discrete approximations. (in preparation).

Rosenthal, J. (1994). Theoretical rates of convergence for Markov chain Monte Carlo. Interface 94.

Tierney, L. (1991). Exploring posterior distributions using Markov chains. *Computing Science and Statistics* **23**, 563–570.

Theoretical rates of convergence for Markov chain Monte Carlo

by

Jeffrey S. Rosenthal*

Department of Statistics, University of Toronto, Toronto, Ontario, Canada M5S 1A1

Phone: (416) 978-4594. Internet: `jeff@utstat.toronto.edu`

(Conference proceedings, Interface '94, June 1994.)

Abstract. We present a general method for proving rigorous, *a priori* bounds on the number of iterations required to achieve convergence of Markov chain Monte Carlo. We describe bounds for specific models of the Gibbs sampler, which have been obtained from the general method. We discuss possibilities for obtaining bounds more generally.

1. Introduction.

Markov chain Monte Carlo techniques, including the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970), data augmentation (Tanner and Wong, 1986), and the Gibbs sampler (Geman and Geman, 1984; Gelfand and Smith, 1990) have become very popular in recent years as a way of generating a sample from complicated probability distributions (such as posterior distributions in Bayesian inference problems). A fundamental issue regarding such techniques is their convergence properties, specifically whether or not the algorithm will converge to the correct distribution, and if so how quickly. Many general convergence results (e.g. Tierney, 1994), qualitative convergence-rate results (Schervish and Carlin, 1992; Liu, Wong, and Kong, 1991a, 1991b; Baxter and Rosenthal, 1994), and convergence diagnostics (e.g. Roberts, 1992; Gelman and Rubin, 1992; Mykland, Tierney, and Yu, 1992) have been developed. However, none of these approaches are entirely satisfactory (Matthews, 1991; Cowles and Carlin, 1994).

In a different direction, a number of papers have attempted to prove rigorous, quantitative bounds on convergence rates for these algorithms (Jerrum and Sinclair, 1989; Frieze, Kannan, and Polson, 1993; Meyn and Tweedie, 1993; Lund and Tweedie, 1993; Mengersen and Tweedie, 1993; Rosenthal, 1991, 1993a, 1993b, 1994). Such results often provide bounds which are very weak, and/or for very specific models, but the area appears to be worthy of further work.

In this paper we describe a general method (Section 2) for proving such quantitative bounds. The method requires only that we verify a drift condition and a minorization condition, for the Markov chain of interest. We describe (Section 3) the application of this (and related) methods to various specific examples of the Gibbs sampler, including variance components models, hierarchical Poisson models, and a model related to James-Stein estimators. In some cases, the bounds appear to be small enough to be of practical use. In other cases, they provide additional theoretical information about the Gibbs sampler for the model being studied.

We close (Section 4) with a brief discussion of possibilities for further bounds of this type.

2. The general method.

The simplest form of our general method is the following, taken from Rosenthal (1993b, Theorem 12).

Proposition. *Let $P(x,\cdot)$ be the transition probabilities for a Markov chain $X_0, X_1, X_2, \ldots$ on a state space $\mathcal{X}$, with stationary distribution $\pi(\cdot)$. Suppose there exist $\epsilon > 0$, $0 < \lambda < 1$, $0 < \Lambda < \infty$, $d > \frac{2\Lambda}{1-\lambda}$, a non-negative function $f : \mathcal{X} \to \mathbf{R}$, and a probability measure $Q(\cdot)$ on $\mathcal{X}$, such that*

$$\mathbf{E}(f(X_1) \,|\, X_0 = x) \;\le\; \lambda f(x) + \Lambda\,, \qquad x \in \mathcal{X} \quad (1)$$

and

$$P(x,\cdot) \;\ge\; \epsilon\, Q(\cdot)\,, \qquad x \in f_d \qquad (2)$$

where $f_d = \{x \in \mathcal{X} \,|\, f(x) \le d\}$, and where $P(x,\cdot) \ge \epsilon\, Q(\cdot)$ means $P(x,S) \ge \epsilon\, Q(S)$ for every measurable $S \subseteq \mathcal{X}$. Then for any $0 < r < 1$, the total variation distance to the stationary distribution after k iterations is bounded above by

$$(1-\epsilon)^{rk} + \left(\alpha^{-(1-r)}\gamma^r\right)^k \left(1 + \frac{\Lambda}{1-\lambda} + \mathbf{E}\left(f(X_0)\right)\right),$$

* Supported in part by NSERC of Canada.

where

$$\alpha^{-1} = \frac{1+2\Lambda+\lambda d}{1+d} < 1; \quad \gamma = 1+2(\lambda d+\Lambda).$$

Inequality (1) above is called a *drift condition*, while inequality (2) above is called a *minorization condition*. The proposition thus allows for precise, quantitative, exponentially-decreasing upper bounds on the distance to stationarity, as a function of the number of iterations k, using just these two inequalities.

The proof of this proposition involves the *coupling inequality*, which states that the total variation distance between the laws of two random variables is bounded by the probability that they are unequal. Proving the proposition thus amounts to (theoretically) constructing auxiliary random variables Y_k, so that $\mathcal{L}(Y_k) = \pi$ but $P(X_k = Y_k)$ is as large as possible. Inequality (2) allows us to construct X_k and Y_k jointly so that, whenever $(X_k, Y_k) \in f_d \times f_d$, they have probability ϵ of becoming equal on the next generation. Furthermore, inequality (1) implies that the number of iterations k for which $(X_k, Y_k) \in f_d \times f_d$ will be large with high probability. Combining these two facts, we can construct X_k and Y_k so that $P(X_k \neq Y_k)$ is small, and thus use the coupling inequality to prove the proposition. The reader is referred to Rosenthal (1993b) for details.

3. Applications to specific models.

The general method of Section 2 (and related methods) have been applied to a number of specific examples of the Gibbs sampler, to derive information about their rates of convergence to the appropriate posterior distributions.

In Rosenthal (1993), a version of the data augmentation algorithm (a special case of the Gibbs sampler) was applied to *finite* sample spaces. It was shown that, with n parameters and n observed data, the algorithm would converge in $O(\log n)$ iterations. Thus, the running time of the algorithm does not grow too quickly with the number of parameters.

In Rosenthal (1991), the Gibbs sampler applied to variance components models (as discussed in Gelfand and Smith, 1990; Gelfand et al., 1990) was analyzed. It was shown that, with K location parameters each having J observed data, the $(K+3)$-dimensional Gibbs sampler would approximately converge in $O\left(1+\frac{\log K}{\log J}\right)$ iterations. So again, the running time of the algorithm does not grow too quickly with the number of parameters.

In Rosenthal (1993b), the Gibbs sampler applied to a hierarchical Poisson model was analyzed, using the same data as analyzed in Gelfand and Smith (1990). For this data, the (11-dimensional) Gibbs sampler was shown to have total variation distance to stationarity after k iterations bounded above by

$$(0.976)^k + (0.951)^k(6.2 + E\left((S^{(0)} - 6.5)^2\right)),$$

where $S^{(0)} = \sum_i \theta_i^{(0)}$ is a sum of initial values. The bound is thus explicit and quantitative, and depends explicitly on the initial distribution. The bound is also not absurdly large: for example, if $E\left((S^{(0)} - 6.5)^2\right) = 2$ and $k = 150$, this bound is equal to 0.03, implying that 150 iterations are sufficient to achieve randomness.

In Rosenthal (1994), the Gibbs sampler applied to a model related to James-Stein estimators (James and Stein, 1961) was analyzed. The model (suggested by Jun Liu) was designed to avoid the use of guesses and empirical estimates in the usual (empirical Bayes) formulation of James-Stein estimators. The Gibbs sampler was intended to facilitate computations related to the associated posterior distribution. A formula was provided which gave a bound on convergence of the Gibbs sampler explicitly, in terms of the number of iterations, the initial distributions, the prior distributions of the model, and the observed data. When applied to the baseball data analyzed in Efron and Morris (1975) and Morris (1983), it proved that the Gibbs sampler would converge in less than 200 iterations.

For certain other prior distributions, it was shown (Rosenthal, 1994) that this Gibbs sampler would in fact not converge at all. This information was used, together with standard convergence theory, to prove that for these (improper) priors, the model itself was improper, i.e. the posterior distribution was non-normalizable. Analysis of the Gibbs sampler was thus used to provide additional information about the model under consideration.

Our method has thus been applied to a variety of realistic examples of the Gibbs sampler. It has provided useful quantitative bounds, convergence information relating the running time to the number of parameters, and additional theoretical information about the underlying model itself.

4. Discussion.

It is now widely recognized that convergence issues are crucial for the successful implementation of Markov chain Monte Carlo algorithms. However, no method is entirely satisfactory for demonstrating such convergence or providing a convergence rate.

We have provided a general method (Section 2) for rigorously and explicitly bounding the convergence of these Markov chain algorithms. The method requires only that we verify a drift condition and a minorization condition for the Markov chain under consideration. In principle the method can be applied to virtually any Markov chain algorithm, and does not require special structure such as spectral information or reversibility. However, it is to be admitted that, in complicated high-dimensional problems, even the verification of the two required conditions can be quite difficult.

We have described the application of this method to several models of the Gibbs sampler. These models are realistic and non-trivial, and our method provides useful information about their convergence properties. The theoretical results appear to be at the point where they can begin to have practical implications.

However, each of these applications has required additional, extensive computation. Furthermore, similar computation may be extremely difficult for more complicated models. Hence, further work is required before these methods are easily usable in very general applied settings. It is possible that the theoretical approach described here can be combined with a more practical analysis, for example by attempting to verify drift and minorization conditions through additional simulation (Cowles and Rosenthal, 1994), which might allow for wider use.

In any case, while there is much work to be done, the methods described here appear to hold promise for providing rigorous rates of convergence for many additional examples of Markov chain Monte Carlo.

Acknowledgements. In the course of this research I have benefited from very helpful conversations with many people, including Persi Diaconis, Jun Liu, Peter Ney, Richard Tweedie, John Baxter, Kate Cowles, and Neal Madras.

REFERENCES

J.R. Baxter and J.S. Rosenthal (1994), Rates of convergence for everywhere-positive Markov chains. Tech. Rep. **9406**, Dept. of Statistics, University of Toronto.

M.K. Cowles and B.P. Carlin (1994), Evaluation and comparison of Markov chain Monte Carlo convergence diagnostics. Tech. Rep., Dept. of Biostatistics, University of Minnesota.

M.K. Cowles and J.S. Rosenthal (1994), work in progress.

B. Efron and C. Morris (1975), Data analysis using Stein's estimator and its generalizations. J. Amer. Stat. Assoc., Vol. **70**, No. **350**, 311-319.

A. Frieze, R. Kannan, and N.G. Polson (1993), Sampling from log-concave distributions. Tech. Rep., School of Computer Science, Carnegie-Mellon University.

A.E. Gelfand and A.F.M. Smith (1990), Sampling based approaches to calculating marginal densities. J. Amer. Stat. Assoc. **85**, 398-409.

A.E. Gelfand, S.E. Hills, A. Racine-Poon, and A.F.M. Smith (1990), Illustration of Bayesian inference in normal data models using Gibbs sampling. J. Amer. Stat. Soc. **85**, 972-985.

A. Gelman and D.B. Rubin (1992), Inference from iterative simulation using multiple sequences. Stat. Sci., Vol. **7**, No. **4**, 457-472.

S. Geman and D. Geman (1984), Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. IEEE Trans. on pattern analysis and machine intelligence **6**, 721-741.

W. James and C. Stein (1961), Estimation with Quadratic Loss. Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Vol. **1**, University of California Press, Berkeley, 361-379.

W.K. Hastings (1970), Monte Carlo sampling methods using Markov chains and their applications. Biometrika **57**, 97-109.

M. Jerrum and A. Sinclair (1989), Approximating the permanent. SIAM J. Comput. **18**, 1149-1178.

J. Liu, W. Wong, and A. Kong (1991a), Correlation structure and the convergence of the Gibbs sampler, *I*. Tech Rep. **299**, Dept. of Statistics, University of Chicago. Biometrika, to appear.

J. Liu, W. Wong, and A. Kong (1991b), Correlation structure and the convergence of the Gibbs sampler, *II*: Applications to various scans. Tech Rep. **304**, Dept. of Statistics, University of Chicago. J. Royal Stat. Sci. **(B)**, to appear.

R.B. Lund and R.L. Tweedie (1993), Geometric convergence rates for stochastically ordered Markov chains. Tech. Rep., Dept. of Statistics, Colorado State University.

P. Matthews (1993), A slowly mixing Markov chain with implications for Gibbs sampling. Stat. Prob. Lett. **17**, 231-236.

K.L. Mengersen and R.L. Tweedie (1993), Rates of convergence of the Hastings and Metropolis algorithms. Tech. Rep. **93/30**, Dept. of Statistics, Colorado State University.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller (1953), Equations of state calculations by fast computing machines. J. Chem. Phys. **21**, 1087-1091.

S.P. Meyn and R.L. Tweedie (1993), Computable bounds for convergence rates of Markov chains. Tech. Rep., Dept. of Statistics, Colorado State University.

C. Morris (1983), Parametric empirical Bayes confidence intervals. Scientific Inference, Data Analysis, and Robustness, 25-50.

P. Mykland, L. Tierney, and B. Yu (1992), Regeneration in Markov chain samplers. Tech. Rep. **585**, School of Statistics, University of Minnesota.

G.O. Roberts (1992), Convergence diagnostics of the Gibbs sampler. In Bayesian Statistics **4** (J.M. Bernardo et al., eds.), 777-784. Oxford University Press.

J.S. Rosenthal (1991), Rates of convergence for Gibbs sampler for variance components models. Tech. Rep. **9322**, Dept. of Statistics, University of Toronto. (Tentatively accepted in *Annals of Statistics.*)

J.S. Rosenthal (1993a), Rates of convergence for Data Augmentation on finite sample spaces. Ann. Appl. Prob., Vol. **3**, No. **3**, 319-339.

J.S. Rosenthal (1993b), Minorization conditions and convergence rates for Markov chain Monte Carlo. Tech. Rep. **9321**, Dept. of Statistics, University of Toronto.

J.S. Rosenthal (1994), Analysis of the Gibbs sampler for a model related to James-Stein estimators. Tech. Rep. **9413**, Dept. of Statistics, University of Toronto.

M.J. Schervish and B.P. Carlin (1992), On the convergence of successive substitution sampling, J. Comp. Graph. Stat. **1**, 111–127.

M.A. Tanner and W.H. Wong (1987), The calculation of posterior distributions by data augmentation (with discussion). J. Amer. Stat. Assoc. **82**, 528-550.

L. Tierney (1994), Markov chains for exploring posterior distributions. Tech. Rep. **560**, School of Statistics, University of Minnesota. Ann. Stat., to appear.

Fraction of Missing Information and Convergence Rate of Data Augmentation *

Jun S. Liu

Department of Statistics, Harvard University, Cambridge, MA 02138

1 Introduction

The Gibbs sampler and other MCMC methods (Gelfand and Smith 1990, Smith and Roberts 1993, Tanner and Wong 1987), which become popular recently in statistical analysis with complicated models, are no more than some devices for generating random samples from an analytically intractable target distribution. The basic idea underlying all these methods is to construct a Markov chain with the target distribution as its equilibrium distribution. The methods differ only in the use of Markov transition functions. For example, the transition function for the Gibbs sampler with systematic scan can be expressed as a product of a sequence of conditional distributions (Smith and Roberts 1993, Liu, Wong and Kong 1994b); while the transition function for a Metropolis-Hastings algorithm consists of a "proposed" transition and a "thinning down" device (Metropolis et al. 1953, Hastings 1970, Smith and Roberts 1993). Many theoretical work has emerged in understanding convergence properties of the MCMC methods. See, for example, Geman and Geman (1984), Gelman and Rubin (1992), Geyer (1992), Liu, et al. (1994a,b), Liu (1992, 1994), Mykland, Tierney and Yu (1993), Roberts (1992), Roberts and Polson (1994), Rosenthal (1993a,b), Schervish and Carlin (1993), Tierney (1991), just to start a list. Here, by taking a slightly different angle to look at the convergence problem, we investigate relationships among various concepts in describing a Gibbs sampler and the associated Bayesian missing data problem: the rate of convergence, sample autocorrelations, and the fraction of missing information.

We distinguish two different situations for the Gibbs sampler: Data Augmentation which refers to a Gibbs sampler with only two iterative components (see Tanner and Wong 1987 for its original version, and Liu et al. 1994a for structural study), and the general Gibbs sampler (Gelfand and Smith 1990). A reason for doing this is that the two component case provides us some extra structure that a general Gibbs sampler does not possess, and the analysis of this simple case can suggest some useful methods for dealing with more general ones.

By making use of covariance structures of Data Augmentation established in Liu et al. (1994a,b), we find that the convergence rate of the induced Markov chain can be characterized by the *maximal fraction of missing information*, which is closely related to the work of Meng and Rubin (1992) for the EM algorithms. Conversely, because of this characterization, we can use autocorrelations of a stationary Gibbs sampling sequence to estimate the fraction of missing information of any quantity of interest, which is useful for deciding how many multiple imputations will be provided.

This article is arranged as follows. We review the concept of fraction of missing information in Section 2. In Section 3, we present structures and several connections for Data Augmentation. A generalization to the general Gibbs sampler is contained in Section 4. A graphical method for comparing different schemes, using the relationships found in Sections 3 and 4, is described in Section 5. In Section 6, we analyze an example for match-making in "broken regression" (DeGroot, Feder, and Goel 1971).

2 The Fraction of Missing Information

The concept of fraction of missing information was first introduced together with the so-called *missing*

* The author thanks Alan Zaslavsky and Yingnian Wu for helpful discussions and computing assistance. This work is partly supported by NSF grant DMS 94-04344 and Milton Fund of Harvard University.

information principle by Orchard and Woodbury (1972). It is later proved to be an important concept for studying the EM algorithms (Dempster, Laird and Rubin 1977). Specifically, Louis (1982) presented a method for finding the observed information, and Meng (1991) and Meng and Rubin (1993) systematically explored the concept and used it to characterize the rate of convergence for the EM and the ECM algorithms.

To introduce the fraction of missing information conveniently, we let Θ denote the parameter vector in our model, let Y denote the observed part of an imaginary complete data set, and let Z denote the missing part. A simple identity underlying the missing information principle and the EM algorithms is

$$\begin{aligned}\log[p(\Theta \mid Y)] &= \log[p(\Theta \mid Y, Z)] \\ &- \log[p(Z \mid \Theta, Y)] + \log[p(Z \mid Y)],\end{aligned}$$

which implies

$$\begin{aligned}-\frac{\partial^2 \log p(\Theta \mid Y)}{\partial\Theta^2} &= -\frac{\partial^2 \log p(\Theta \mid Y, Z)}{\partial\Theta^2} \\ &+\frac{\partial^2 \log p(Z \mid \Theta, Y)}{\partial\Theta^2}.\end{aligned}$$

Integrating out the missing data Z with respect to $p(Z|\Theta, Y)$, we arrive at the following missing information principle

$$\begin{aligned}\text{Observed Information} &= \text{Complete Information} \\ &- \text{Missing Information.}\end{aligned}$$

Denoting each term by I_{obs}, I_{com}, and I_{mis}, respectively, we can define the *fraction of missing information* as

$$\gamma_L = \frac{I_{mis}(\Theta)}{I_{com}(\Theta)} = 1 - \frac{I_{obs}(\Theta)}{I_{com}(\Theta)},$$

where the I functions are evaluated at the true parameter value. When Θ is a 1-dim parameter, the above quantity is well defined. Otherwise, the above definition takes a matrix form. Meng (1991) used the largest eigenvalue of the missing fraction matrix $I_{mis}^{-1}(\Theta)I_{com}(\Theta)$ to characterize the convergence rate of the EM algorithm.

Now let us take a Bayesian viewpoint. Suppose a prior distribution $p_0(\Theta)$ is given, and we are interested in $h \equiv h(\Theta)$ (one can view this as a way of eliminating nuisance parameters). If one can impute the missing data, i.e., draw samples $Z^{(1)}, \ldots, Z^{(m)}$ from the predictive distribution $p(Z|Y)$, then the posterior distribution of h, $p(h|Y)$, can be approximated by

$$p(h \mid y) \approx \frac{1}{m}\{p(h|Y, Z^{(1)}) + \cdots p(h|Y, Z^{(m)})\}.$$

For example, $Z^{(1)}, \cdots, Z^{(m)}$ can be draws from an iterative sampling scheme. When using the above multiple imputation type of approximations, *the fraction of missing information* is usually important for one to understand the impact of the missing data on the estimation of h. Also, it is important for one to decide how many imputations should be provided. As Rubin (1987) advocated, m can be chosen as small as 3 to 5 for estimating posterior mean of h. Of course, in this case, the fraction of missing information with respect to h can not be too high.

The fraction of missing information in the Bayesian framework can be easily defined as (Rubin 1987)

$$\begin{aligned}\gamma_B &= \frac{\text{var}\{E(h \mid Y, Z) \mid Y\}}{\text{var}(h \mid Y)} \\ &= 1 - \frac{E\{\text{var}(h \mid Y, Z) \mid Y\}}{\text{var}(h \mid Y)}\end{aligned}$$

which can be explained as the extra variation caused by missing Z.

Note that in large sample and when $h=\theta$, since $\text{var}(h|Y) \approx 1/I_{obs}$ and $E\{\text{var}(h|Y, Z)\} \approx 1/I_{com}$, the two definitions of the fraction of missing information, γ_B and γ_L, are equivalent.

3 Structures for Data Augmentation

We call a special situation of the Gibbs sampler *Data Augmentation* if there are only two components for iterative sampling (Liu et al. 1994). We use Θ and Z to denote the respective components in Data Augmentation to emphasize its connection with Bayesian missing data problems.

Let $\Theta^{(1)}, Z^{(1)}, \Theta^{(2)}, Z^{(2)}, \ldots,$ be consecutive draws from a stationary Data Augmentation. In other words, we assume that $\Theta^{(1)}$ is drawn from the target distribution $p(\Theta|Y, Z)$. In the following, since everything is conditioned on Y, we will omit it in all expressions. For example, when we write $E\{h(\Theta)|Z\}$, it actually means $E\{h(\Theta)|Y, Z\}$.

Consider two consecutive draws from Data Augmentation, we find that

$$E(h^{(k)}h^{(k+1)}) = E\{E(h^{(k)}h^{(k+1)} \mid Z^{(k)})\} \qquad (1)$$

$$\begin{aligned} &= E\{E(h^{(k)} \mid Z^{(k)})E(h^{(k+1)} \mid Z^{(k)})\} \\ &= E\{E^2(h \mid Z)\}, \end{aligned}$$

where the first equality follows from an elementary fact that $E(A) = E[E(A|B)]$; the second and third equalities follow from the fact that $\Theta^{(k)}$ and $\Theta^{(k+1)}$ are conditionally independent and identically distributed given $Z^{(k)}$. These facts can be illustrated by the following diagram:

$$\begin{tikzcd}
\Theta^{(1)} \arrow[dr] & & \Theta^{(2)} \arrow[dr] & & \Theta^{(3)} \arrow[dr] & & \cdots \\
& Z^{(1)} \arrow[ur] & & Z^{(2)} \arrow[ur] & & Z^{(3)} & \cdots
\end{tikzcd}$$

From the diagram, we observe that $\Theta^{(1)}$ connects with $\Theta^{(2)}$ through $Z^{(1)}$, and, from the definition of the scheme, $(\Theta^{(1)}, Z^{(1)})$ and $(\Theta^{(2)}, Z^{(1)})$ have the same joint distribution when the chain is stationary. These two properties only hold for Data Augmentation, not for the general Gibbs sampler. However, this type of dependence graph can be applied to a general Gibbs sampler and provide useful intuition. In Section 5 we will illustrate how to use these diagrams to compare different schemes.

As a consequence of (2), we have the following identity

$$\mathrm{cov}\{h(\Theta^{(k)}), h(\Theta^{(k+1)})\} = \mathrm{var}[E\{h(\Theta) \mid Z\}]$$

The formula implies that the correlation coefficient between the two consecutive h's are

$$\rho(h^{(k)}, h^{(k+1)}) = \gamma_{_B}.$$

An intuition of this is that the higher the fraction of missing information, the more "sticky" the sample outputs from Data Augmentation, and vice versa. The extra variance caused by the missing data, $\mathrm{var}\{E(h|Z)\}$, can then be estimated as

$$\hat{v}_{mis} = \frac{1}{m-1}\sum_{k=1}^{m-1} h^{(k)}h^{(k+1)} - (\bar{h}_m)^2.$$

If, on the other hand, $g(Z) = E(h|Z)$ is easy to compute, one may also approximate $\mathrm{var}\{E(h|Z)\}$ by

$$\tilde{v}_{mis} = \sum_{i=1}^{m}(g^{(i)} - \bar{g}_m)^2/(m-1),$$

where $g^{(i)} = E(h|Z^{(i)})$ and $\bar{g}_m = (g^{(1)} + \cdots + g^{(m)})/m$. This is a variation of Rao-Blackwellization (Gelfand and Smith 1990, Liu et al. 1994a).

Intuitively, it seems that the latter estimation is better. For example,

$$\begin{aligned} \mathrm{var}\{h^{(1)}h^{(2)}\} &= E\{(h^{(1)}h^{(2)})^2\} - [E\{E^2(h|Z)\}]^2 \\ &= E\{E^2(h^2 \mid Z)\} - [E\{E^2(h|Z)\}]^2, \end{aligned}$$

while

$$\mathrm{var}(g^2) = E\{E^4(h \mid Z)\} - [E\{E^2(h \mid Z)\}]^2.$$

Hence, by the Cauchy-Schwarz inequality, we have

$$\mathrm{var}(g^2) \leq \mathrm{var}\{h^{(1)}h^{(2)}\}.$$

Furthermore, by Theorem 3.1 of Liu et al. (1994)

$$\begin{aligned} &\mathrm{cov}\{(g^{(1)})^2, (g^{(k+1)})^2\} \\ = \;& \mathrm{var}\{E(\cdots E[E\{g^2(Z)|\Theta\}|Z]\cdots)\} \end{aligned}$$

where the right hand side has k expectation signs. Also, we notice that

$$E\{g^2(Z)|\Theta\} = E\{E[g(Z)h(\Theta)|Z]|\Theta\}.$$

For $\hat{v}_{mis}$, we let $f(\Theta) = E[E\{h(\Theta)|Z\}|\Theta]$, which is just $E(h^{(2)}|\Theta^{(1)})$. Then we have

$$\begin{aligned} &\mathrm{cov}(h^{(1)}h^{(2)}, h^{(k+1)}h^{(k+2)}) \\ = \;& \mathrm{cov}(h^{(2)}f^{(2)}, h^{(k+1)}h^{(k+2)}) \end{aligned}$$

which, for the same reason as above, has the following expression

$$\mathrm{var}\{E(\cdots E[E\{h(\Theta)f(\Theta)|Z\}|\Theta]\cdots)\}$$

where there are $k-1$ expectation signs on the right hand side. However

$$E\{h(\Theta)f(\Theta)|Z\} = E\{E[h(\Theta)g(Z) \mid \Theta] \mid Z\}$$

If we compare the expression of lag-k autocovariance for the $(g^{(i)})^2$ sequence with that for the $h^{(i)}h^{(i+1)}$ sequence, we find that the former always has one more conditional expectation sign than the latter. However since the orders of the conditionings are different, there is no clear comparison between the two except for the case when lag=1, in which case, the autocovariance for the latter expression is always greater than or equal to the former.

The following analogy is helpful for understanding the above discussion. Consider two scenarios: (i) a vector **a** is projected to vector **b** and then to vector **c**; (ii) **a** is directly projected to **c**. How do we compare the length of the projections? Apparently, if the three vectors are in the same plane and **b** lies between **a** and **c**, the latter projection is smaller than the former one. But in most other cases, the former is smaller than the latter. This corresponds to comparing $\text{var}[E\{E(X|Y)|Z\}]$ and $\text{var}\{E(X|Z)\}$.

For any two random variables U and V, we define the *maximal correlation* between them as

$$R(U,V) = \sup_{\text{var}\{t(U)\}=\text{var}\{s(V)\}=1} corr\{t(U), s(V)\}.$$

It is well understood that for a reversible stationary Markov chain $X^{(1)}, X^{(2)}, \ldots$, the maximal correlations between two consecutive states, $R(X^{(k)}, X^{(k+1)})$, is equal to λ, where $1-\lambda$ is the so-called "spectral gap." See Liu et al. (1994a,b) for more references. For discrete case, λ is just the magnitude of the second largest eigenvalue (in absolute value). For nonreversible chain, the scaled long-range maximal correlation is equal to λ (Liu et a. 1994b). That is,

$$\lim_{k\to\infty} \{R(X^{(1)}, X^{(k+1)})\}^{1/k} = \lambda.$$

It is shown in Liu et al. (1994a) that the maximal correlation between two consecutive draws of Data Augmentation, $R(\Theta^{(k)}, \Theta^{(k+1)})$ is the intrinsic rate of convergence of the scheme, and is equal to $R^2(\Theta, Z)$.

On the other hand, under mild conditions (see Csàki and Fischer 1960), there exists a pair of functions $h_0(\Theta)$ and $g_0(Z)$ with unit variance such that $\text{corr}(h_0, g_0)=R(\Theta, Z)$ (denoted as R later), and

$$E\{g_0(Z) \mid \Theta\} = R\, h_0(\Theta) \quad (2)$$

$$E\{h_0(\Theta) \mid Z\} = R\, g_0(Z) \quad (3)$$

Therefore, h_0 suffers the *maximal fraction of missing information*

$$\gamma_B(h_0) = \text{var}\{E(h_0|Z)\}/\text{var}(h_0) = R^2,$$

and the maximal fraction of missing information is equal to the rate of convergence of Data Augmentation. If a function h is correlated with h_0 (with respect to π), then

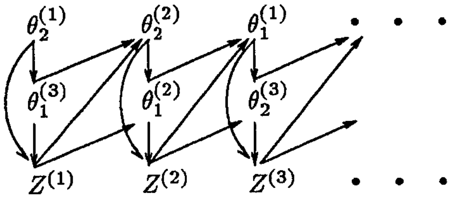
$$\{corr(h^{(1)}, h^{(k+1)})\}^{1/k} \to \lambda$$

as k goes to infinity. This follows from spectral decomposition of h (Liu 1991, Garen and Smith 1994, Roberts 1992). It suggests that the *maximal fraction of missing information* can be estimated by the output sequence of the Gibbs sampler.

4 Missing Information in the General Gibbs Sampler

We now turn our attention to the general Gibbs sampler with systematic scan. There are two situations commonly encountered in practice. We shall discuss them in the order of increasing complexity.

Case 1. $\Theta = (\theta_1, \theta_2)$, $Z = Z$. That is, given Θ, Z can be drawn directly; but θ_1 must be drawn conditional on both θ_2 and Z, and θ_2 must be drawn conditional on θ_1 and Z. Note that this can be generalized obviously. The following diagram illustrates the sampler:



Hence,

$$\begin{aligned} &\text{cov}\{h(\theta_1^{(1)}), h(\theta_1^{(2)})\} \\ = \;& E\{h(\theta_1^{(1)})h(\theta_1^{(2)})\} - E\{h(\theta_1)^2\} \\ = \;& \text{var}[E\{h(\theta_1) \mid \theta_2, Z\}] \end{aligned}$$

which implies that lag-1 autocorrelation of the h sequence is in general not its fraction of missing information with respect to Z, but is a quantity that reflects dependency between θ_1 and (θ_2, Z). Note that

$$\text{var}[E\{h(\theta_1) \mid \theta_2, Z\}] \geq \text{var}[E\{h(\theta_1)|Z\}].$$

Another way around is to design a function $g(Z)$ and to estimate the maximal correlation between Θ and Z from it. For example, if it happens that we know g_0 in (2) and (3), then by Lemma 4 of Liu (1994),

$$\begin{aligned} \text{cov}\{g_0(Z^{(k)}), g_0(Z^{(k+1)})\} &= \text{var}[E\{g_0(Z) \mid \Theta\}] \\ &= R^2 \, \text{var}\{h_0(\Theta)\}. \end{aligned}$$

Here R^2 is the maximal fraction of missing information and is an upper bound for $\gamma_B(h)$. This duality provides us the following scheme for obtaining an estimate of the maximal fraction of missing information.

Step 1. Design a function $g(Z)$. Usually this can just be a linear function (e.g., see Liu 1991).

Step 2. Estimate lag-k autocorrelation r_k for the g sequence for $k = 1, 2, \ldots$, after the chain converges, and fit the exponential model

$$r_k = c\rho^k.$$

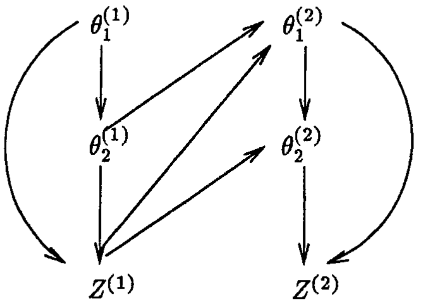
Garren and Smith (1994) provided refined methods. The fitted value $\hat{\rho}$ is an estimate of $R(\Theta, Z)$.

Case 2. $\Theta = (\theta_1, \theta_2)$ and $Z = (z_1, z_2)$. This is the case where the fraction of missing information can not be estimated from the sample autocorrelations. The maximal fraction of missing information can be extracted from long range autocorrelations by the same reason as explained in Case 1.

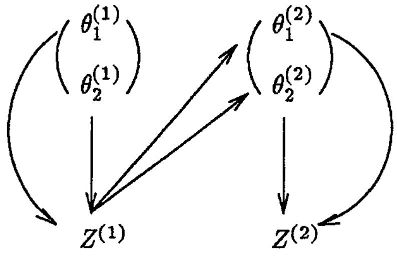
5 Compare Schemes via Diagrams

In running a Gibbs sampler or a more general MCMC algorithm, one usually has flexibilities in designing sampling schemes. As with many iterative methods , we are usually faced with a dilemma: we either have to sacrifice computational ease for iterative simulation in exchange for fast convergence, or have to suffer slow convergence in exchange for computational simplicity. Only in some rare situations as explored in Liu (1994) be we satisfied in both ways. Specifically, when the Bayesian predictive distribution is simple, one can use the ***predictive updated*** version to improve convergence without sacrificing computational simplicity. Liu et al. (1994a) and Liu (1994) provided some theoretical arguments based on operator theory. Here we use diagrams to illustrate autocorrelation structures. We hope that the analysis in this section can shed light on more complicated general situations.

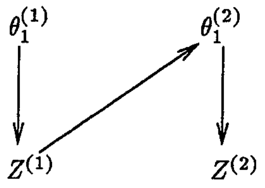
For the sake of simple argument, suppose the sampler involves three components (θ_1, θ_2, Z) and each component is visited in turn: $\theta_1 \rightarrow \theta_2 \rightarrow Z$. The following diagram shows dependency between two consecutive iterations. For example, $\theta_1^{(2)}$ is generated by a draw from $\pi(\theta_1|\theta_2^{(1)}, Z)$, which is illustrated in the diagram by two arrows connecting $\theta_2^{(1)}$ and Z with $\theta_1^{(2)}$. Other arrows have similar implications. This diagram shows that the two consecutive states depend on each other via the connection between $(\theta_1^{(1)}, Z^{(1)})$ and $(\theta_1^{(2)}, \theta_2^{(2)})$ as illustrated by three arrows in the middle or the diagram.



Next diagram illustrates a ***grouping*** scheme, where it is assumed that given Z, (θ_1, θ_2) can be drawn together. The diagram illustrates that dependency between two consecutive states is via the connection between $Z^{(1)}$ and $(\theta_1^{(2)}, \theta_2^{(2)})$, where only two arrows are used for this connection. Compared with the above diagram for the original sampler, dependency between the two consecutive states for ***grouping*** is weaker.



Our final diagram represents the ***collapsing*** scheme, in which we assume that θ_2 can be theoretically integrated out so that the sampler is applied only to the two remaining components. In this diagram, the only connection between two consecutive states is that between $Z^{(1)}$ and $\theta_1^{(2)}$. Only one arrow is used, which indicates the weakest correlation among the three schemes.



We expect that this type of analysis can be generalized to other situations to help one design efficient sampling schemes.

6 An Example: Broken Regression

Suppose x_i, $i = 1, \ldots, 100$, are i.i.d. normal with variance τ^2; and $y_i = \alpha + \beta x_i + \epsilon_i$, where the ϵ_i are i.i.d. from $N(0, \sigma^2)$. It is a standard regression problem if we observe (x_i, y_i) for $i = 1, \cdots, 100$. Suppose, however, the pairing information is somehow lost and we can only observe u_i, $i = 1, \ldots, 100$, a random shuffle of the y_i. The problem is no longer trivial. This can also be viewed as a special case of file matching problem. DeGroot et al. (1971) studied this problem with an objective to maximize the number of correct matches. We are interested in estimating β and the corresponding fraction of missing information (for not knowing the matching).

Let Q be the permutation that produces the u_i from the y_i. The main difficulty is that Q is missing. Let $\Theta = (\alpha, \beta)$ and $U = (u_1, \ldots, u_{100})$. With a prior distribution on Θ, Data Augmentation can be applied if we can (a) draw Q from $p(Q|\Theta, U)$ and (b) draw Θ from $p(\Theta|Q, U)$. Step (b) is simple since it only involves multivariate t-distribution. Step (a) is nontrivial. As was implemented in a preliminary report of Y. Wu (Dept. of Statist., Harvard U.), step (a) can be accommodated by a "Metropolized shuffling" scheme. Roughly speaking, a random shuffling scheme is employed that provides us a Markov chain on the space of all permutations. Based on this chain, we can apply Metropolis-Hastings rejection rule to achieve our target distribution $p(Q|\Theta, U)$. In our simulation, we used switch shuffling (randomly draw two cards and switch them). Within each iteration (i.e., a cycle of Steps (a) and (b)), 500 Metropolized shuffles were conducted, since, as theory suggested, $O(n \log(n))$ steps are needed to shuffle n cards uniformly.

We simulated a data set with $\tau^2 = 1$, $\sigma^2 = 1$, and $\alpha = 0$. Assuming that $\alpha = 0$ is known, we used a flat prior for β. Figure 1 illustrates our results. Panel(1,1) shows the posterior distribution of β, where the x's were simulated from $N(0, 1)$ and the true β was zero. As indicated, its variance is 0.12, considerably larger than 0.01, the complete-data posterior variance of β. Panel(1,2) shows the autocorrelations among the β's. The fraction of missing information can be estimated as $\hat{\gamma}_B = 0.924$ from the autocorrelation plot. As theory in Sections 2 and 3 indicated,

$$(1 - \gamma_B)\text{var}(\beta \mid U) = E\{\text{var}(\beta \mid U, Q)\}$$

where the RHS is average complete-data variance. This identity was experimentally confirmed since $(1 - 0.923) \times 0.12 = 0.009$ which is close to the theoretical value 0.01. Panel(2,1) is the same posterior distribution, but the x's were simulated from $N(0, 1)$ and the true β=0. With the x's far from origin, both the posterior variance, 0.021, and the fraction of missing information, 0.619, were considerably smaller. In Panel(3,1), the x were simulated from $N(1, 1)$ and the true $\beta = 1$. It seems to suggest that the fraction of missing information is not related to the true value of β, but is very sensitive to $\sum x_i^2$.

An intuitive solution of the problem is to sort both the x and the u first and then do a regression on the sorted data. But this procedure overestimates β and does not provide proper inference. The above Bayesian method we employed, however, is unbiased (with flat prior) and supplies proper variance estimation. When $\sum x_i^2$ is extremely large, the sorting method (essentially any method) works well, implying that the matching information is unimportant for the inference of β. This, together with the foregoing simulation study, suggests a conjecture that the fraction of missing information for β monotonely decreases as $\sum x_i^2$ increases.

References

Csàki, P., and Fischer, J.H.(1960), "Contributions to the problem of maximal correlation," *Matematikao Kotato Intezet, Kozlemenyei*, **5**, 325-337.

DeGroot, M.H., Feder, P.I., and Goel, P.K. (1971), "Matchmaking," *Ann. Math. Statist.*, **42**, 578-593.

Dempster, A.P., Laird, N., and Rubin, D.B. (1977), "Maximum likelihood from incomplete data via the EM algorithm (with discussion)," *J. Roy. Statist. Soc.*, Ser. B, **39**, 1-38.

Gelfand, A.E. and Smith, A.F.M. (1990), "Sampling-based approaches to calculating marginal densities," *J. Amer. Statist. Assoc.*, **85**, 398–409.

Gelman, A. and Rubin, D.B. (1992), "Inference from iterative simulation using multiple sequences (with discussion)," *Statist. Sci.*, **7**, 457-511.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. on Pattn Anal. and Mach. Intell.*, **6**, 721–741.

Geyer, C.J. (1992), "Practical Markov chain Monte Carlo", *Statist. Sci.*, **7**, 473-483.

Hastings, W.K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, **57**, 97-109.

Liu, J.S. (1991), "Correlation structure and convergence rate of the Gibbs sampler," Ph.D. *Thesis*, Dept. of Statist., U. of Chicago.

Liu, J.S. (1992), "Metropolized independent sampling scheme with comparisons to rejection sampling and importance sampling," *Tech. Rep.*, Stat. Dept., Harvard U. To appear in *Statistics and Computing.*

Liu, J.S. (1994), "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *J. Amer. Statist. Assoc.*, **89**, in press.

Liu, J.S., Wong, W.H. and Kong, A. (1994a), "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," *Biometrika*, **81**, 27-40.

Liu, J.S., Wong, W.H. and Kong, A. (1994b), "Covariance structure and convergence rate of the Gibbs Sampler with various scans", *J. Roy. Statist. Soc.*, Ser. B **55**, in press.

Meng, X. (1991), "Towards complete results for some incomplete-data problems," Ph.D. *Thesis*, Dept. of Statist., Harvard U.

Meng, X. and Rubin, D.B. (1993), "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, 267-278.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953), "Equations of state calculations by fast computing machines," *J. Chem. Phys.*, **21**, 1087-1091.

Mykland, P., Tierney, L. and Yu, B. (1992), "Regeneration in Markov chain samplers," *Tech. Rep.*, Dept of Statist., U. of Chicago.

Orchard, T. and Woodbury, M.A. (1972), "A missing information principle: theory and applications," In *Proc. of the 6th Berkeley Symposium on Math. Stat. and Prob.*, 697-715.

Roberts, G.O. and Polson, N.G. (1994), "On the geometric convergence of the Gibbs sampler," *J. Roy. Statist. Soc.*, Ser. B **55**, 377-384.

Roberts, G.O. (1992), "Convergence diagnostics of the Gibbs sampler," In *Bayesian Statistics* **4**, eds. J. Bernardo, J.O. berger, A.P. Dawid and A.F.M. Smith, 763-773, Oxford University Press.

Rosenthal, J.S. (1993a), "Rates of convergence for Data Augmentation on finite sample spaces," *Ann. Appl. Prob.* **3**, 319-339.

Rosenthal, J.S. (1993b), "Minorization conditions and convergence rates for Markov chain Monte Carlo," *Tech. Rep.*, Dept. of Statist., U. of Toronto.

Rubin, D.B. (1987), *Multiple Imputations for Nonresponse in Surveys.* Wiley, New York.

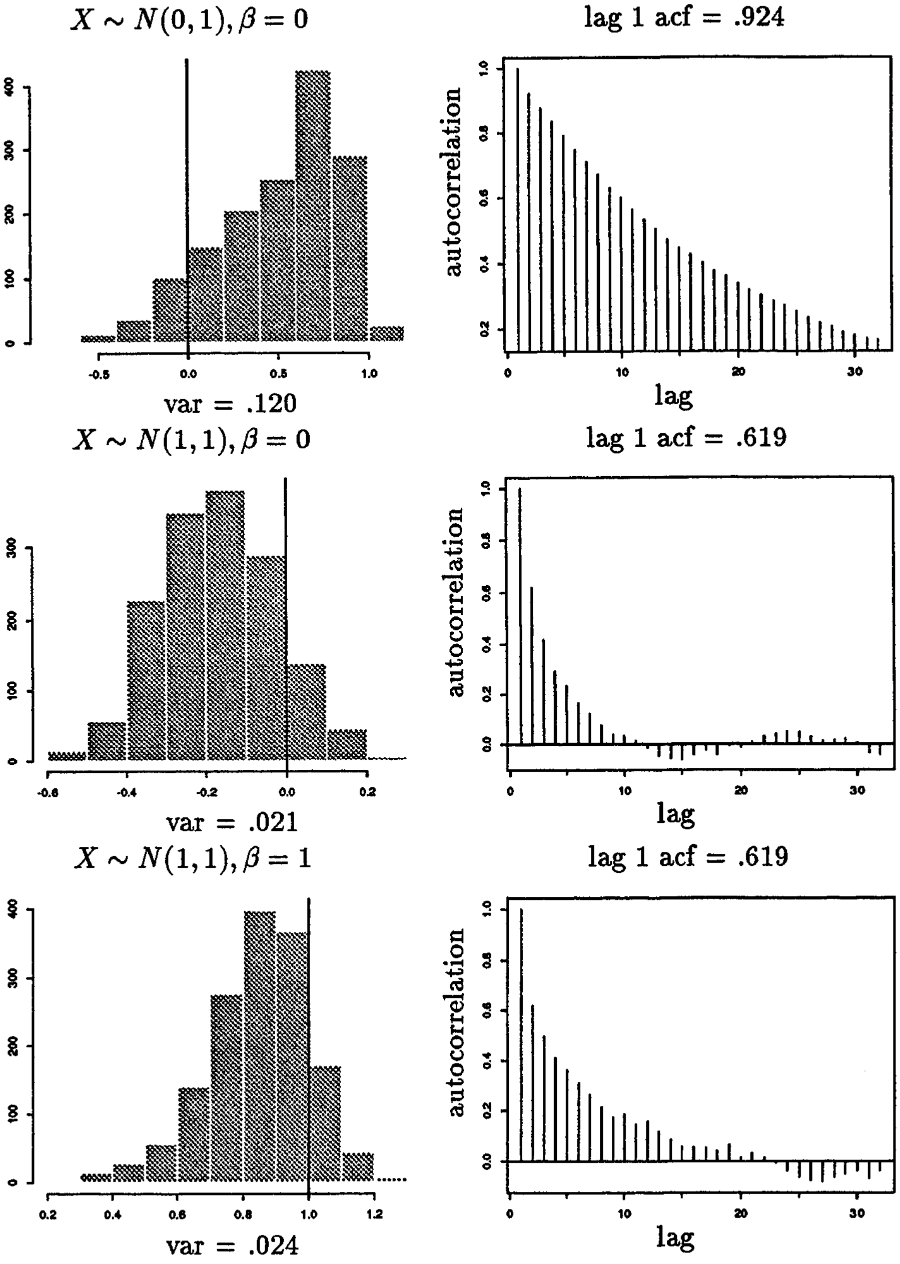
Schervish, M.J., and Carlin, B.P. (1992), "On the convergence of successive substitution sampling," *J. Comp. Graph. Statist.* **1**, 111-127.

Garren, S.T. and Smith, R.L. (1994), "Convergence diagnostics for Markov chain samplers," *Tech. Rep.*, Dept. of Statist., U. of North Carolina.

Smith, A.F.M., and Roberts, G.O. (1993), "Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods (with discussion)," *J. Roy. Statist. Soc.*, Ser. B, **55**, 3-23.

Tanner, M.A. and Wong, W.H. (1987), "The calculation of posterior distributions by data augmentation (with discussion)," *J. Amer. Statist. Assoc.*, **82**, 528–550.

Tierney, L. (1991), "Markov chains for exploring posterior distributions", *Tech. Rep. 560*, School of Statistics, University of Minnesota.



$X \sim N(0,1), \beta = 0$

lag 1 acf = .924

var = .120

$X \sim N(1,1), \beta = 0$

lag 1 acf = .619

var = .021

$X \sim N(1,1), \beta = 1$

lag 1 acf = .619

var = .024

Figure 1: Results for the broken regression

Monte Carlo Estimation of Multilocus Autozygosity Probabilities.

Elizabeth Thompson*

Department of Statistics, GN-22,
University of Washington
Seattle, WA 98195

Abstract

The ability to sample latent variables using Markov chain Monte Carlo (MCMC) has had a major impact on computations relating to the genetic analysis of complex traits, or traits observed on complex pedigrees. One area in which exact likelihood computation is often infeasible is multilocus linkage mapping. One method of linkage analysis for rare recessive traits is homozygosity mapping where data on affected inbred individuals are analysed. Key to this method are the patterns of autozygosity in the individuals, and MCMC provides also a method for studying these patterns. Algorithms for the exact computation of autozygosity probabilities on an arbitrary pedigree very rapidly become computationally infeasible. However, an MCMC algorithm can provide accurate estimates in reasonable computing time, and these probabilities can then be used to map the genes responsible for disease.

1. Introduction

Monte Carlo likelihood is becoming increasingly used where exact likelihood analysis is computationally infeasible. One area in which such likelihoods arise is that of genetic mapping, where the locations in the genome of genes influencing a given trait are to be inferred.

The elements of genetic models are straightforward: genes exist, genes segregate (are copied) from parents to offspring, and the types of genes carried by an individual influence observable trait characteristics. A locus is a specification of the position of a gene on a chromosome. With modern molecular genetic techniques, individuals can be typed for a wide variety of DNA markers of known location in the genome. These DNA markers can be chosen to be highly polymorphic; there are many different alleles (types of genes) that an individual may have. The genes at these DNA marker loci segregate in a Mendelian way (Mendel, 1866); each individual has two genes at the locus, one a copy of a randomly chosen one of the two in his father, and the other a copy of a randomly chosen one in his mother. Segregation of genes from different parents to a child, and from a parent to different children, are independent. These simple 50/50 probabilities underlie all of genetics, but in considering the joint segregation at several genetic loci, or the pattern of single-locus segregations on an extended family, computations can rapidly become very complex, principally because not all the relevant information can be observed.

Genetic loci, $L_1, ..., L_k$ that index segments of DNA on the same chromosome are "linked"; the segregations of genes at two loci are not independent. If the maternal gene at locus L_h in a father segregates to a child, it is more probable that the gene that segregates at an adjacent locus, L_j, is also the father's maternal gene. Similarly for the father's paternal gene, and similarly also for genes segregating from the mother. This dependence can be expressed through the "recombination fractions", $r_{h,j}$, between the two loci. Specifically, the probability that genes at loci L_h and L_j segregating from one parent to the child have different grandparental origins is $r_{h,j}$. In fact, the value of a recombination fraction between two loci depends on numerous factors, most importantly on the sex of the parent. This fact can be incorporated into analyses, but, for simplicity, is ignored in the current paper.

The biological phenomenon underlying recombination is a "crossover" between the two parental chromosomes in the formation of the offspring chromosome. There will be a recombination between loci L_h and L_j if there is an odd number of crossover events. The genetic (map) distance between two loci is the expected number of crossovers between them, and hence is additive (Haldane, 1919). However, the data provide information only on recombination frequencies between loci (Fisher, 1922). This pattern is related to map distance, but also

*Research supported in part by PHS grant GM-46255.

depends on the pattern of interference. Interference is the name given to the biological phenomenon that a crossover at one point on a chromosome affects the chance that crossovers occur at other points in the vicinity. Under an assumption of no interference, recombination events occur along the chromosome as a Poisson process rate 1, when the chromosome is measured in units of map distance. In practice, interference exists, particularly where the loci are close together and recombination fractions between them are small. However, the amount of data required to estimate levels and patterns of interference seldom exists in human genetic studies. In genetic mapping, the objective is to detect linkage, to infer locus order, and place loci on a chromosome by estimating recombination fractions between them. For such purposes, interference can safely be ignored.

Now in mapping a genetic disease, marker types will be available for some individuals in a pedigree in which the disease is segregating. Disease or relevant quantitative trait data will be available also for some members of the pedigree. However, first, not all individuals will be observed; some will be unavailable, particularly ancestors. Second, the genes underlying the trait phenotypes may not be determined; for example, for a recessive disease, two copies of the disease allele are needed to express the trait, but those who do not express it may have one copy of the disease allele, or none. Third, even where single-locus marker genotypes are observable the haplotype information is not; that is, it is not known which alleles are on the same chromosome, having been received from the same parent. One set of single-locus genotypes (a specification of the unordered pair of alleles at each locus) can correspond to many different multilocus genotypes (a specification of the alleles on each chromosome, at each of the loci). Thus in computing a likelihood, for a given locus order and set of recombination fractions, a huge sum over all the possible configurations of haplotypes is required. With the increasing availability of DNA markers there is an increasing potential for mapping traits with more limited trait data or more complex modes of expression. However, more markers, and marker loci with more alleles, and traits observable for a more limited subset of the pedigree members, all compound the computational difficulties, since the number of possible underlying configurations of genes on all the relevant members of the pedigree increases vastly.

Thus, with the increasing desire to examine multiple markers, and markers with multiple alleles, a major limitation of linkage analysis has become the practical and theoretical bounds on the computational feasibility of likelihood evaluation. There are many further aspects of linkage analysis, and many alternative approaches to localising the genes responsible for a genetic disease. A much fuller description of standard statistical methods in linkage analysis may be found in the text by Ott (1991).

In this paper, we consider one possible approach to the computations needed to map a rare recessive disease from data on affected inbred individuals. We consider only marker loci at which the types of the two genes carried by a observed individual are known, and a recessive disease for which it is known whether or not an observed individual carries two copies of the disease allele. The (multilocus) genotype G_i of individual i is a specification of the types of the genes on each of a pair of chromosomes of the individual. The phenotype Y_i of i is a specification of the observed trait characteristics determined by the underlying genotypes. We subsume all the parameters of the genetic model into the parameter vector θ, and use $P_\theta(\cdot)$ to denote probabilities under the model. The total set of genotypes on a pedigree is denoted $\mathbf{G}$, and of observed phenotypes $\mathbf{Y}$.

2. Monte Carlo likelihood

Monte Carlo estimates of integrals or expectations are not new, either in general (Hammersley and Handscomb, 1964) or in genetic linkage analysis (Thompson et al. 1978). However, Monte Carlo methods have only become widely used with the explosion in use of Markov chain Monte Carlo (MCMC) which permits simulation from distributions known only up to a normalising constant, and hence simulation from conditional distributions. The statistical problems involved in fitting genetic linkage models to trait data, $\mathbf{Y}$, on a set of related individuals may be viewed as latent variable or "missing data" problems. Were all the underlying genetic events observable, likelihood computation and parameter estimation would be trivial, but only trait data (phenotypes) of some individuals are observed. We denote the latent variables by $\mathbf{X}$.

The likelihood is

$$L(\theta) = P_\theta(\mathbf{Y}) = \sum_{\mathbf{X}} P_\theta(\mathbf{Y}, \mathbf{X}) = \sum_{\mathbf{X}} P_\theta(\mathbf{Y}|\mathbf{X}) P_\theta(\mathbf{X}) \quad (1)$$

Although the summation may be infeasible, we suppose that the latent variables, $\mathbf{X}$, are chosen in such a way that each term of the expression is easily computed.

Monte Carlo estimators of likelihood ratios can be based on

$$\frac{L(\theta)}{L(\theta_0)} = \frac{P_\theta(\mathbf{Y})}{P_{\theta_0}(\mathbf{Y})} = \mathrm{E}_{\theta_0}\left(\frac{P_\theta(\mathbf{Y},\mathbf{X})}{P_{\theta_0}(\mathbf{Y},\mathbf{X})} \;\middle|\; \mathbf{Y}\right) \quad (2)$$

(Thompson and Guo, 1991), provided simulation from the appropriate distribution is possible. Suppose $\mathbf{X}(l)$,

$l = 1, ..., N$, are realisations from $P_{\theta_0}(\mathbf{X}|\mathbf{Y})$ then a Monte Carlo estimate of the likelihood ratio (2) is

$$\frac{1}{N}\sum_{l=1}^{N}\left(\frac{P_\theta(\mathbf{Y},\mathbf{X}(l))}{P_{\theta_0}(\mathbf{Y},\mathbf{X}(l))}\right) \tag{3}$$

From an importance sampling perspective, the estimator (3) is efficient; for values of θ_0 close to θ the sampling distribution mimics the shape of the integrand $P_\theta(\mathbf{Y},\mathbf{X})$ of (1). Further, equation (3), through simulation at a given θ_0, provides a likelihood ratio approximant, as a function of θ, in the sense of Geyer and Thompson (1992). At least for values of θ close to θ_0, a single simulation provides an estimate of the local likelihood surface.

In Monte Carlo approaches to complex problems with many latent variables, the key is simulation conditional upon data; that is from

$$P_{\theta_0}(\mathbf{X} \mid \mathbf{Y}) \;=\; P_{\theta_0}(\mathbf{X},\mathbf{Y})/P_{\theta_0}(\mathbf{Y}) \tag{4}$$

With well chosen latent variables $\mathbf{X}$, the numerator of this expression is readily evaluated, but the denominator is

$$L(\theta_0) = P_{\theta_0}(\mathbf{Y}) = \sum_{\mathbf{X}} P_{\theta_0}(\mathbf{X},\mathbf{Y})$$

and this summation is often infeasible. The denominator is, in fact, precisely the likelihood whose exact evaluation is often impossible, necessitating the Monte Carlo estimation.

Metropolis-Hastings algorithms are Markov chain Monte Carlo methods designed to meet this need, providing realisations (approximately) from a distribution known up to a normalising constant (Hastings, 1970). For each $\mathbf{X}$ a "proposal distribution" $q(\cdot,\mathbf{X})$ is defined. Then, if the process is now at $\mathbf{X}$ the next value is generated as follows:

1. Generate $\mathbf{X}^*$ from the proposal distribution $q(\cdot,\mathbf{X})$
2. Compute the Hastings ratio

$$\lambda = \frac{q(\mathbf{X},\mathbf{X}^*)P_{\theta_0}(\mathbf{X}^*|\mathbf{Y})}{q(\mathbf{X}^*,\mathbf{X})P_{\theta_0}(\mathbf{X}|\mathbf{Y})} = \frac{q(\mathbf{X},\mathbf{X}^*)P_{\theta_0}(\mathbf{Y},\mathbf{X}^*)}{q(\mathbf{X}^*,\mathbf{X})P_{\theta_0}(\mathbf{Y},\mathbf{X})}$$

Note that λ can be computed without knowledge of $P_{\theta_0}(\mathbf{Y})$.

3. With probability $\lambda^* = min(1,\lambda)$ the process moves to $\mathbf{X}^*$ and with probability $(1-\lambda^*)$ it remains at $\mathbf{X}$.

The distribution (4) is an equilibrium distribution of the Markov chain just defined. Provided $q(\cdot,\cdot)$ is chosen so that the chain is ergodic, running the chain provides (after a sufficient number of steps for convergence) realisations from the distribution (4). The algorithm of Metropolis et al. (1953) is a special case; if $q(\mathbf{X}^*,\mathbf{X}) = q(\mathbf{X},\mathbf{X}^*)$ the Hastings ratio reduces to the odds ratio of the proposal state $\mathbf{X}^*$ versus the current state $\mathbf{X}$.

In the genetic context, the latent variables $\mathbf{X}$ have normally been taken to be the underlying multilocus genotypes (the pairs of haplotypes) carried by each individual in the pedigree. This makes for easy evaluation of $P_{\theta_0}(\mathbf{X},\mathbf{Y})$ but not for easy sampling of the large space of possible $\mathbf{X}$-values. The space of Lange and Matthysse (1989) is even larger, including also indicators of the grandparental origins of genes. Although local updating methods are very slow, they are convenient for genetic analysis problems. If large changes in genotypic configuration are proposed, the Hastings ratio can be impossible to compute, and constraints in the feasible genotypic patterns on pedigrees mean that almost all proposals have zero probability.

There are various approaches to improving sampler performance in genetic problems. Lin (1993) made great progress towards increasing the practicality of MCMC methods in linkage analysis, using Metropolis-coupled samplers (Geyer, 1991), and a form of "heating" in the Metropolis-Hastings steps to improve mixing of the chain. Geyer and Thompson (1994) used simulated tempering (Marinari and Parisi, 1992) to make sampling feasible on a very large complex pedigree with many constraints. These strategies result in a sampler that can sample genotypes efficiently on a large pedigree. However, for several linked markers, the huge space of possible genotypic configurations that then arises may render the sampler ineffective.

An alternative approach is to consider alternative latent variables $\mathbf{X}$, to produce a smaller space more easily sampled by MCMC methods. Note that the requirements on $\mathbf{X}$ are only that $P_\theta(\mathbf{Y},\mathbf{X})$ should be very quickly computable. Now $P_\theta(\mathbf{Y},\mathbf{X})$ is normally computed as $P_\theta(\mathbf{Y} \mid \mathbf{X})P_\theta(\mathbf{X})$. Thus any $\mathbf{X}$ for which these two factors can be readily computed will suffice. For the problem of mapping rare recessive traits from data on inbred affected individuals, it is possible to bypass the multilocus genotypes of unobserved individuals, and use only segregation indicators as the latent variables.

3. Homozygosity mapping.

In linkage analysis, due to uncertainties as to whether an unaffected individual carries a disease gene, the computational difficulties on extended pedigrees, and the costs of typing large numbers of individuals, there have been many approaches towards basing linkage analyses on a small number of observed (usually affected) individuals. The extreme case is ***homozygosity mapping*** in which a rare recessive is mapped using only marker and trait data on independent inbred affected individuals.

It was first pointed out by Smith (1953), that individuals affected with rare recessive diseases provide information for linkage analysis, even without any marker or phenotype data on other relatives. For a recessive disease, affected individuals are *homozygous* at the disease locus; that is, they carry two copies of the same allele. For a rare disease, many affected individuals are so through being the offspring of consanguineous marriages, and thus receiving two copies of the disease gene identical-by-descent or *autozygous* from a recent common ancestor of the two parents. In this case, the affected individual is likely to be homozygous also at closely linked markers, and this homozygosity provides evidence for linkage. Unrelated inbred individuals will be homozygous at independent segments of the genome, but the shared affected status of the individuals will cause shared homozygosity in the neighbourhood of the disease locus. The scope of homozygosity mapping, which is simply linkage analysis using data only on unrelated inbred affected individuals, was extended by Lander and Botstein (1987). With a dense map of highly polymorphic DNA markers, a small number of affected individuals can provide substantial information for mapping a recessive disease gene.

Linkage analysis is the analysis of cosegregation of genes at different loci, from parents to offspring. If two loci are tightly linked, there is a high probability that if the individual receives a grandmaternal [grandpaternal] allele from his mother at one locus, he will do so also at the adjacent one, and similarly for the gene received from his father. The key underlying events that determine the data on the affected inbred individual are the segregations that specify the ancestral genes that he receives. Let m and p index the maternal and paternal segregations to some individual. Let $S_{mj} = 0$ if the maternal allele received by the individual at locus j is of grandmaternal origin, and $S_{mj} = 1$ otherwise, and let S_{pj} be similarly defined for the paternal allele. Then, at any locus j,

$$\begin{aligned} P(S_{mj}=0) &= P(S_{mj}=1) = \\ P(S_{pj}=0) &= P(S_{pj}=1) = \frac{1}{2} \end{aligned}$$

and at two loci h and j

$$P(S_{mh}=S_{mj}) = P(S_{ph}=S_{pj}) = (1-r_{h,j})$$

where $r_{h,j}$ is the recombination fraction between the two loci.

Then for a given segregation i, the recombination events are determined by segregation indicators S_{ij}, $j = 1, ..., k$, where S_{ij} is 0 or 1 as the origin of the segregating gene at locus j is grandmaternal or grandpaternal, respectively. That is, we shall take the indicators $\mathbf{S} = \{S_{ij}\}$ as the latent variables $\mathbf{X}$ in the Monte Carlo likelihood framework of section 2. Figure 1 shows the case of the offspring of a first-cousin marriage. At any locus, the offspring individual may receive genes autozygous from either of his parents' common grandparents; there are eight relevant segregation indicators that will specify the gene descents.

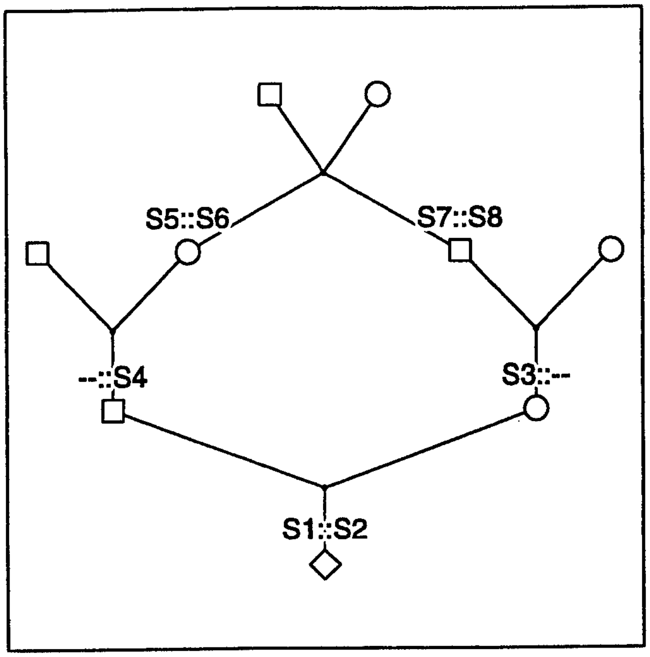


Figure 1: A first cousin marriage, showing segregation indicators.

Table 1: Example of segregation array, for the pedigree of figure 1

Segreg.:	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Locus								
L_1	0	1	1	1	0	1	0	0
L_2	0	1	1	0	0	1	0	0
L_3	0	1	1	0	0	1	0	1
L_4	0	1	0	0	0	1	0	1

Table 1 shows four successive patterns of values for the eight segregation indicators of figure 1, such as might arise along a chromosome segment, or at four loci. In the first pattern, the paternal gene of the offspring individual derives from his grandmother ($S_1 = 0$), and is the paternal gene of this grandmother ($S_4 = 1$), and is in fact the great-grandfather's maternal gene ($S_5 = 0$). Likewise the final individual's maternal gene is this same maternal gene in his great-grandfather; the individual is autozygous for this gene. By locus 2, S_4 has become 0; the final individual's paternal gene is now the paternal ($S_6 = 1$) gene of his great-grandmother. By locus 3,

S_8 has become 1; this leaves the genes in the final individual unchanged, since $S_3 = 1$, so the grandfather's maternal gene is not transmitted. However, by locus 4, S_3 becomes 0; now the final individual is autozygous for the paternal gene in his great-grandmother.

Consider now Table 1 as illustrating a possible value of the state **S** at four loci being used in a genetic analysis. The prior probabilities of **S** are straightforward. However, for implementation of a Metropolis algorithm, relative values of $P_{\theta_0}(\mathbf{Y}, \mathbf{S})$ are required, or $P_{\theta_0}(\mathbf{Y} \mid \mathbf{S})$. The binary indicators, $\mathbf{S} = \{S_{ij}\}$, of grandparental origins of genes in each given offspring individual, at each locus readily determine the multilocus autozygosity patterns in the observed individual. This is done simply by following the descent paths of genes, as in the example described above; an efficient algorithm is easily implemented to update these descent paths, and hence the resulting autozygosity pattern, when a S_{ij} changes. For a single observed individual, the autozygosity pattern is k binary indicators, specifying whether or not the S_{ij} result in the individual having two genes autozygous at locus j, $j = 1, ..., k$. The probability of a genotype homozygous for an allele with frequency q is q^2 or q, as the individual is not/is autozygous at the locus. The probability of a heterozygous genotype is 0 if the individual is autozygous at the locus, and is $2q_1q_2$ otherwise, where q_1 and q_2 are the two allele frequencies.

Table 2: Probability ratios of segregation indicators S_{ij}

$S_{i,j-1}$	$S_{i,j+1}$	*probability ratio**
1	1	$(1-r_{j-1})(1-r_j)/r_{j-1}r_j$
1	0	$(1-r_{j-1})r_j/r_{j-1}(1-r_j)$
0	1	$r_{j-1}(1-r_j)/(1-r_{j-1})r_j$
0	0	$r_{j-1}r_j/(1-r_{j-1})(1-r_j)$

* : $P(S_{ij} = 1 \mid \mathbf{S}_{-(ij)})/P(S_{ij} = 0 \mid \mathbf{S}_{-(ij)})$.
r_h is the recombination frequency between L_h and L_{h+1}.
Note also:
$P(S_{ij} = 1 \mid \mathbf{S}_{-(ij)}) = P(S_{ij} = 1 \mid S_{i,j-1}, S_{i,j+1})$
$P(S_{ij} = 0 \mid \mathbf{S}_{-(ij)}) = P(S_{ij} = 0 \mid S_{i,j-1}, S_{i,j+1})$
$\mathbf{S}_{-(ij)}$ denotes all elements of **S** other than S_{ij}.

The space of **S**-values is also easy to sample from. The simplest algorithm uses a Metropolis proposals to change the grandparental origin of the gene at a random locus in a random segregation. The probability ratio for the proposed change in **S** depends only on the indicators at adjacent loci for the same segregation (Table 2). For example suppose the current **S** were that of Table 1, and the proposal was to change $S_{4,2}$ from its current value 0 to 1. This would eliminate a recombination between loci 1 and 2 ($S_{4,1} = 1$) giving probability ratio

$$(1 - r_{1,2})/r_{1,2},$$

but create one between loci 2 and 3 ($S_{4,3} = 0$) giving another factor

$$r_{2,3}/(1 - r_{2,3}).$$

This recombination ratio is then weighted by the appropriate conditional probability of phenotypic observations $P_{\theta_0}(\mathbf{Y} \mid \mathbf{S})$, for current and proposed **S**-values. This sampler is clearly irreducible: if a given pattern of autozygosity in the observed individual is compatible with the data, then so also is any pattern with fewer loci at which the affected individual is autozygous and hence homozygous.

Werner's syndrome (WS) is a very rare recessive genetic disease of premature aging. It has recently been mapped to chromosome 8 using outbred affected relatives (Goto et al., 1992), and this linkage has been confirmed by analysis of a set of inbred affected individuals (Schellenberg et al., 1992) in 21 small pedigrees of Japanese and Caucasian origin. The frequency of the disease allele is assumed to be 0.004. A Monte Carlo linkage likelihood analysis of a subset of five of these pedigrees is given by Thompson (1994); here we use just two of the pedigrees for purposes of illustration. Two markers were of significance in the published linkage reports: $D8S87$ and ANK. Originally ANK and $D8S87$ were thought to be flanking markers, but the likely order is now thought to be (WS, $D8S87$, ANK). For the purposes of illustration only, we take the recombination fractions between WS and $D8S87$ and between $D8S87$ and ANK each to be 0.1; this is probably larger than the true values, but of the correct order of magnitude. Data and information on these markers were provided by Dr. Ellen Wijsman (personal communication).

4. Autozygosity probabilities

In fact, for a single affected inbred individual, the data **Y** at a position h on a chromosome depend on $\mathbf{S}(h)$ only through $Z(h)$, the autozygosity (I) or non-autozygosity (N) in the inbred affected individual. Over multiple loci, or along the chromosome continuum, these patterns of autozygosity are themselves of interest. Although, for a very rare recessive trait, the posterior probability of autozygosity at the disease locus is very high, the probability that all of a set of unrelated affected inbred individuals are autozygous may be low. Further, the way in which such posterior probabilities are influenced by data on linked markers is non-trivial, for the patterns of autozygosity along a chromosome segment follow no simple process. Specifically, even in the absence of interference, the process is not Markov, since it is an

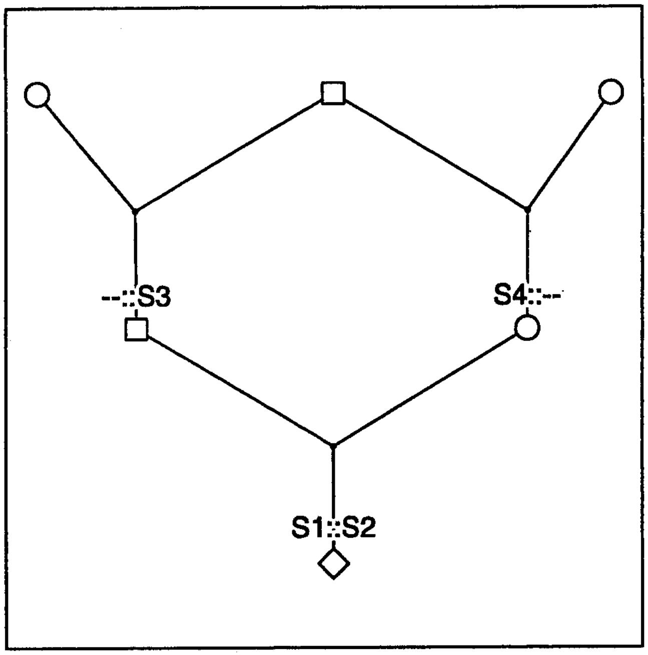


Figure 2: A half-sib marriage example.

aggregate process and shows the clumping phenomenon typical of such processes (Aldous, 1989; Blossey, 1993).

Consider first the prior, disregarding data **Y**. The smallest non-trivial example consists of the offspring of a half-sib mating (figure 2). This is also the largest example for which the space of **S**-values can be drawn readily (figure 3). As one moves along the chromosome, the process $\mathbf{S}(h)$ performs a random walk at rate n on the vertices of the n-dimensional hypercube (Donnelly, 1983). Here, $n = 4$ and, without loss of generality, the two vertices positioned as shown in figure 3 are those which result in autozygosity of the inbred offspring individual: $\mathbf{Z}(h) = I$ if $S_1(h) = S_2(h) = 1$ and $S_3(h) = S_4(h)$. Overall, $P(Z(h) = I) = 2/16 = 0.125$. When $Z(h) = I$, the next jump of the random walk will require $Z(h) = N$; when $Z(h) = N$, the next jump results in $Z(h) = I$ with overall probability $(2 \times \frac{1}{2} + 4 \times \frac{1}{4})/14 = 1/7$. However, although by symmetry $Z(h) = I$ is a renewal point of the process, when the process leaves $Z(h) = I$, the probability that the next jump will result in a return to $Z(h) = I$ is 3/8. The overall probability, $P(Z(h) = I)$ can easily be computed on even a complex pedigree; it is simply the inbreeding coefficient of the individual. For two loci, at given recombination fraction, the probability of autozygosity at both loci can be computed by the algorithm of Thompson (1988), again even on a complex pedigree. However, due to the non-Markov pattern of autozygosity along the chromosome, these marginal and pairwise probabilities do not suffice.

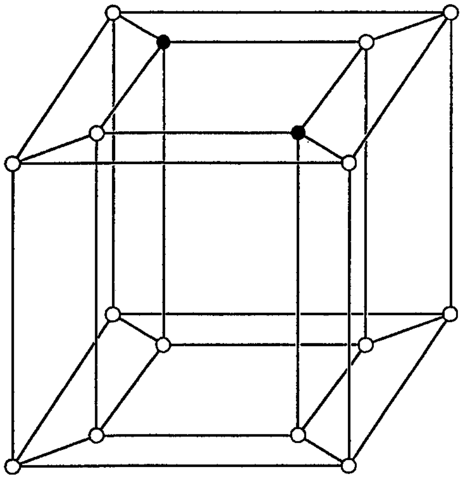


Figure 3: Random walk structure, corresponding to figure 2.

Table 3: Prior autozygosity probabilities for cousin marriage.

state Z			*True*(1)	*Markov*(2)
N	N	N	0.8825	0.8811
N	N	I	0.0264	0.0277
N	I	N	0.0131	0.0131
N	I	I	0.0155	0.0155
I	N	N	0.0264	0.0277
I	N	I	0.0022	0.0009
I	I	N	0.0155	0.0155
I	I	I	0.0184	0.0183

(1) Results from 10^9 MCMC steps and 10^8 i.i.d realisations are almost identical to 10^{-4}.
(2) Results from assuming (incorrectly) a first-order Markov chain for autozygosity at successive loci.

Table 3 shows the autozygosity probabilities for three loci, with recombination fraction 0.1 between each pair of adjacent loci, for the case of a first-cousin marriage (figure 1). For this small problem, exact results could have been obtained, but in fact these are Monte Carlo results, obtained both by 10^9 Metropolis steps of MCMC, and also by 10^8 independent realisations from the prior. For this problem, these two simulations give comparable accuracy (to $\pm 10^{-4}$) in comparable computing time (about 8 hours on a DEC3100). Also shown are the probabilities that would be given by a first-order Markov process with the same pairwise and marginal probabilities. First, it can be seen that $Z = I$ is a renewal point; where the central locus has $Z = I$ the "Markov" results agree with the correct results. Second, the major effect, in terms of relative error, is in the case $\mathbf{Z} = (I, N, I)$; there is a clumping of states $Z = I$ in the jump chain. Alternatively viewed, there is an increased probability of small regions of non-autozygosity (and hence likely heterozygosity at a highly polymorphic marker) within regions of autozygosity (and hence homozygosity). In this example, the sequence (I, N, I) has probability 2.5 times larger than a "Markov" view would predict.

Table 4: Prior autozygosity probabilities for pedigree of figure 4.

state	**Z**		$True^{(1)}$	$Markov^{(2)}$
N	N	N	0.7901	0.7889
N	N	I	0.0478	0.0493
N	I	N	0.0257	0.0251
N	I	I	0.0271	0.0273
I	N	N	0.0478	0.0493
I	N	I	0.0050	0.0031
I	I	N	0.0271	0.0273
I	I	I	0.0295	0.0297

(1) Results from 10^9 MCMC steps and 10^8 i.i.d realisations are almost identical to 10^{-4}
(2) Results from assuming (incorrectly) a first-order Markov chain for autozygosity at successive loci.

Another example is given in Table 4. Many of the pedigrees in the Werner's syndrome data set are first cousin marriages. The more complex pedigree (figure 4) was first ascertained as a first cousin marriage, but later it was discovered that each parent of the affected proband was also the offspring of a first cousin marriage, as shown. Although this is a small pedigree, exact linkage likelihood computations become infeasible with the standard methods with more than three loci, due to the pedigree complexity. The final offspring individual can be autozygous for a gene in any of the three original founders marked. Again, the "true" results in Table 4 are Monte Carlo results (both 10^8 independent samples and 10^9 MCMC steps, agreeing to 4 decimal places). The "Markov" assumption again underestimates most severely the probability of $\mathbf{Z} = (I, N, I)$. However, note also that now there is no renewal when $Z = I$; the lack of symmetry of the three relevant founder ancestors destroys this property, even though numerically the discrepancies are small.

Generally, for just three loci, only the low-probability state $\mathbf{Z} = (I, N, I)$ shows substantial departure from the first-order Markov probability values. However, with data, this state may have high posterior probability. One of the first cousin marriages (figure 1) for the Werner's syndrome (WS) data illustrates this. The data consist of homozygosity (affected) at the WS locus (allele frequency 0.004), heterozygosity at the marker locus $D8S87$ (for two alleles, each frequency 0.5) and homozygosity at the ANK marker. The allele at ANK has population frequency 0.44, so homozygosity is not strong evidence of autozygosity, but the example will serve.

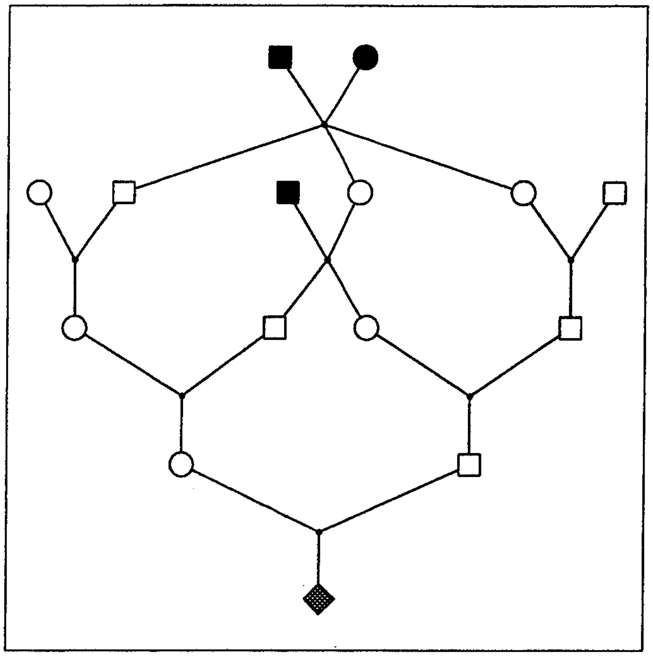


Figure 4: A more complex pedigree.

Table 5: Posterior autozygosity probabilities for cousin marriage.

state	**Z**		$MCMC^{(1)}$	$ratio^{(2)}$
N	N	N	0.1001	0.1134
N	N	I	0.0068	0.2576
I	N	N	0.7491	28.3750
I	N	I	0.1440	65.4545

(1) Results 10^9 MCMC steps, agree with *prior* × *likelihood* to within standard error.
(2) Ratio of posterior to prior probability (see text).

Table 5 shows the posterior probabilities of the four relevant autozygosity states; states autozygous at the $D8S87$ locus are eliminated by the data, and so not listed. As expected, the states with autozygosity at the WS locus have much increased probability *a posteriori*;

the WS disease allele has population frequency only 0.004. Note in particular that the state with lowest prior probability now has a probability 0.1440, 65 times higher than before. Of course, the posterior probabilities could also be obtained by multiplying the prior state probabilities by the likelihoods (and this was done as a check). Prior state probabilities can be efficiently obtained by i.i.d Monte Carlo, but conditional probabilities can only be sampled via MCMC. However, even in this simple example, the standard error of the MCMC estimate for the state INI is smaller, for an equal amount of computing time, due to the 65-fold factor between prior and posterior. When, as here, the range of the ratios of posterior to prior is 3 orders of magnitude, sampling from the prior, and using importance sampling to reweight to the posterior, is far less efficient than sampling from the posterior, even though the latter requires use of MCMC.

5. Discussion

Monte Carlo estimation provides an approach when exact likelihood and probability computation is infeasible, particularly in problems of complex dependent highly structured data, such as arise in genetic analysis. There are many ways to set up the Markov chain Monte Carlo likelihood estimates via a choice of latent variables. In this paper, we have focussed on one particular choice – the use of segregartion indicators. This seems to have promise in cases where a very few individuals are observed on each of a number of possibly large pedigrees, the individuals being observed for a number of DNA markers. A particular case is homozygosity mapping, where the key is the posterior pattern of autozygosity (gene identity by descent) in affected inbred individuals.

MCMC is used to sample from posterior distributions, but this does not require a Bayesian analysis. Realisations from the distribution of latent variables, conditional on the data, but at prespecified parameter values, can be used to provide efficient Monte Carlo estimates of a likelihood surface. Moreover, while multilocus genotypes are key unobservables in genetic analysis, it may not always be efficient to consider these the latent variables in a Monte Carlo analysis; segregation indicators that specify the passage of genes segregating in a pedigree are more fundamental even than genotypes, and, provided the relevant probabilities of observed data given the latent variables can be easily computed, the genotypes of individuals can be bypassed.

Autozygosity patterns at multiple linked loci become of increasing relevance as multilocus linkage analyses are performed. The random walk framework of Donnelly (1983), and the Posson clumping heuristic of Aldous (1989) together make study of the prior probability distribution of patterns more feasible (Blossey, 1993). However, in order to assess autozygosity in the light of data, or to use realisations from the posterior distribution of autozygosity consitional on data in a likelihood analysis, MCMC provides the most efficient computational approach in many cases. Posterior probabilities of autozygosity patterns are more efficiently estimated by MCMC, than by reweighting prior probabilities estimated by i.i.d Monte Carlo.

Acknowledgement

I am grateful to Ellen Wijsman, for discussion on homozygosity mapping and Werner's syndrome.

References

Aldous, D. J. (1989). *Probability approximations via the Poisson clumping heuristic* Springer; Berlin.

Blossey, H. (1993). *The Poisson clumping heuristic and the survival of genome in small pedigrees* Ph.D. Thesis, University of Washington.

Donnelly, K. P. (1983). The probability that related individuals share some section of the genome identical by descent. *Theoretical Population Biology* **23** 34–64.

Fisher, R. A. (1922). The systematic location of genes by means of crossover observations. *American Naturalist* **56** 406–411.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface.* Pp 156–163. Interface Foundation of North America.

Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data (with Discussion). *Journal of the Royal Statistical Society (B)* **54** 657–699.

Geyer, C. J. and Thompson, E. A. (1994). Annealing Markov chain Monte Carlo with applications to pedigree analysis. *Submitted*

Goto, M., Rubenstein, M., Weber, J., Woods, K., and Drayna, D. (1992). Genetic linkage of Werner's syndrome to five markers on chromosome 8. *Nature* **355** 735–738.

Haldane, J. B. S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics* **8** 299-309.

Hammersley, J. M., and Handscomb, D. C. (1964). *Monte Carlo Methods.* Methuen & Co., London.

Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

Lander, E. S., and Botstein, D. (1987). Homozygosity mapping: a way to map human recessive traits with the DNA of inbred children. *Science* **236** 1567-1570.

Lange, K., and Matthysse, S. (1989). Simulation of pedigree genotypes by random walks. *American Journal of Human Genetics* **45** 959–970.

Lin, S. (1993). *Markov chain Monte Carlo estimates of probabilities on complex structures.* Ph.D. Thesis, University of Washington.

Marinari, E. and Parisi G. (1992). Simulated tempering: a new Monte Carlo scheme. *Europhys. Lett.* **19** 451–458.

Mendel, G. (1866). *Experiments in Plant Hybridisation.* Mendel's original paper in English translation, with a commentary by R.A. Fisher: Oliver and Boyd, Edinburgh, 1965.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of State Calculations by Fast Computing Machines. *Journal of Chemical Physics* **21** 1087–1092.

Ott, J. (1991). *Analysis of Human Genetic Linkage.* 2 nd. edition. The Johns Hopkins University Press; Baltimore, MD.

Schellenberg, G. D., Martin, G. M., Wijsman, E. M., Nakura, J., Miki, T., and Ogihara, T. (1992). Homozygosity mapping and Werner's syndrome. *The Lancet* **339** 1002.

Smith, C. A. B. (1953). Detection of linkage in human genetics. *Journal of the Royal Statistical Society (B)* **15** 153–192.

Thompson, E. A., Kravitz, K., Hill, J., and Skolnick, M. H. (1978). Linkage and the power of a pedigree structure. In *Genetic Epidemiology* N.E. Morton (ed.), pp. 247–253. Academic Press, New York.

Thompson, E. A. (1988). Two-locus and three-locus gene identity by descent in pedigrees. *IMA Journal of Mathematics Applied in Medicine & Biology* **5** 261–280.

Thompson E. A and Guo, S.-W. (1991). Monte Carlo evaluation of likelihood ratios. *IMA Journal of Mathematics Applied in Medicine & Biology* **8** 149-169.

Thompson, E. A. (1994). Monte Carlo likelihood in genetic mapping. *Statistical Science:* in press.

Statistical and Computational Challenges in Physical Mapping

David O. Nelson[1]
Lawrence Livermore National Laboratory
Statistics Department, U.C. Berkeley

Terry Speed[2]
Statistics Department, U.C. Berkeley

Abstract

One of the great success stories of modern molecular genetics has been the ability of biologists to isolate and characterize the genes responsible for serious inherited diseases like Huntington's disease, cystic fibrosis, and myotonic dystrophy. Instrumental in these efforts has been the construction of so-called "physical maps" of large regions of human chromosomes.

Constructing a physical map of a chromosome presents a number of interesting challenges to the computational statistician. In addition to the general ill-posedness of the problem, complications include the size of the data sets, computational complexity, and the pervasiveness of experimental error. The nature of the problem and the presence of many levels of experimental uncertainty make statistical approaches to map construction appealing. Simultaneously, however, the size and combinatorial complexity of the problem make such approaches computationally demanding.

In this paper we discuss what physical maps are and describe three different kinds of physical maps, outlining issues which arise in constructing them. In addition, we describe our experience with powerful, interactive statistical computing environments. We found that the ability to create high-level specifications of proposed algorithms which could then be directly executed provided a flexible rapid prototyping facility for developing new statistical models and methods. The ability to check the implementation of an algorithm by comparing its results to that of an executable specification enabled us to rapidly debug both specification and implementation in an environment of changing needs.

1 Overview

One major goal of the Human Genome Project (Olson 1993) is to reduce the time and expense required to isolate and study regions of biological interest by constructing physical maps of the entire human genome. Such maps can then be used by other molecular biologists involved in the interesting and difficult task of understanding how the approximately 100,000 genes buried in our chromosomes conspire to make us human beings.

In this article we will concentrate on issues involved in constructing physical maps. First, we will describe what physical maps are. Then we will discuss some of the statistical and computational problems associated with constructing various kinds of physical maps, specifically

- STS content maps,
- maps based on random fingerprinting, and
- restriction maps.

Finally, we will describe how we used a modern, statistical computing environment to help us with the tricky task of ensuring that the programs we implemented were faithful to the ideas and algorithms we designed.

As an aside, the reader should be aware that biology is one of those sciences where exceptions and special cases abound: nearly every general statement one can make turns out to be wrong. Physical mapping is certainly no exception to this situation. In the interests of clarity and brevity, however, we will confine our attention only to typical examples and refrain from the impulse to be general or encyclopedic. For a more thorough introduction, see Nelson and Speed (1994).

[1] Research was performed under the auspices of the U. S. Department of Energy by Lawrence Livermore National Laboratory under contract number W-7405-ENG-48, with additional support from NSF grant DMS-91-13527.

[2] Research was partially supported by NSF grant DMS-91-13527.

2 What is a "physical map?"

What are physical maps? The answer is not as precise as one would like. To understand this, we must first understand something about recombinant DNA techniques as well as current limitations in how regions of DNA can be analyzed by molecular geneticists. (See Brown (1990) for a readable introduction to recombinant DNA techniques and genetic analysis.) A fundamental problem in molecular genetics is that

- current methods of chemically analyzing substantial stretches of DNA require a sample containing a large number of identical molecules, typically produced by recombinant DNA amplification; however
- the maximum size of a region that can be amplified by current techniques is orders of magnitude smaller than even the smallest human chromosome.

For example, the size of the longest contiguous fragment of DNA that can be reliably amplified by a recombinant DNA process called "cloning" ranges from around 4×10^4 to 1×10^6, depending on the vector and host. Similarly, the longest stretch of DNA that can be reliably amplified by a purely chemical technique known as polymerase chain reaction (PCR) is approximately 1×10^3 bases. In contrast, the twenty-two human autosomes range in size from around 3×10^8 bases for chromosome 1 down to about 5×10^7 bases for chromosome 21. Because of this mismatch in sizes, producing enough DNA to permit biochemical analyses currently requires a process called *cloning*, in which

- a large number of identical chromosomes are broken randomly into fragments by one or more of a class of enzymes known historically as *restriction enzymes*,
- individual fragments of appropriate size are incorporated by biological or chemical mechanisms into the DNA of host organisms such as *E. coli* or yeast,
- the individual hosts are separated from each other and allowed to grow in into colonies, with the fragment in each host being replicated along with the DNA of the host during cell division (*mitosis)*.

In this way, the natural DNA replication machinery of the host organism is exploited to replicate the fragment along with the host's chromosomes. After enough mitoses, each host colony can be harvested. The result of this process is a *library* of cloned chromosome fragments, where each fragment is present in large enough quantities to permit isolation and purification of the fragment and subsequent biochemical analyses. Unfortunately, the library contains no information about the relative positions of the fragments along the chromosome. *Physical maps* are data structures which provide the necessary information to enable the order and distance among fragments to be deduced. Hence, they are essential if a collection of overlapping cloned chromosome fragments (a *contig*) is to be treated as though it were a contiguous region of DNA.

Outside of the genetics community, the process of physical mapping is much less well known than the process of *genetic mapping*, as described by Elizabeth Thompson (this proceedings). Table 1 on the following page attempts to clarify the situation by contrasting several attributes of the two types of maps. In both cases, one is attempting to detect relationships and compute "distances" between genetic objects of interest. In genetic mapping, one uses data from pedigrees and phenotypes to estimate the expected number of recombinations between two loci of interest.

In physical mapping, on the other hand, one uses data from experiments which we call "fingerprints" to determine order and distance between clones or more abstract objects called *sequence-tagged sites* (STSs), which we will define presently. In this context, a "fingerprint" for a clone consists of data from one or more experiments on that clone, the results of which depend in some way on the underlying DNA sequence. Hence, the results of these experiments can help identify or characterize the clone. Cloned fragments which overlap, i.e., share a portion of the genome, *may* produce fingerprints more similar to one another than clones which do not overlap.

	Genetic Mapping	**Physical Mapping**
Objects	Genes or loci	Clones or STSs
Distance	Expected number of recombinations	Base pairs
Data	Pedigrees and phenotypes	"Fingerprints"
Goal	Order and distance among genes or loci	Order and Distance among clones or STSs
Why?	Localize gene to small region of genome	Prepare for biochemistry: sequencing, probing ...

Table 1: A comparison of genetic and physical mapping.

			Physical	**Genetic Distances**	
Ch 19 Marker Pairs			**Distance**	**Female**	**Male**
D19S20	→	D19S247	1.5 Mb	9.4 cM	30.5 cM
D19S177	→	D19S76	2.0 Mb	6.1 cM	4.6 cM
D19S76	→	D19S179	12.0 Mb	19.1 cM	10.7 cM

Table 2: A comparison of genetic and physical distances.

Since genetic maps provide information on distances between loci, and loci can often be associated with clones, one might wonder why geneticists don't just use the genetic map to determine distances between clones. Table 2 show why. It describes physical and genetic distances between four polymorphic markers on chromosome 19. These four markers span a region from a point near the end of the short arm of the chromosome (D19S20) down to a point near the center of the chromosome (D19S179).

One immediately sees from this table that physical distance is only loosely correlated with genetic distance. What is more, genetic distances are sex-specific. Typically, many more recombinations occur in sperm than in eggs. However, as can be seen from the three pairs in Table 2, this is not always the case. Consequently, although genetic distances are used as rough guides to physical distances (the rule-of-thumb is 10^6 bases per centimorgan), this correspondence is rough indeed, and physical maps must be constructed to determine the precise physical relationships among genetic objects.

3 Constructing Physical Maps

Now, let us turn to the process of constructing physical maps to see what roles computers and statistics play. In this section, we will describe how three different kinds of maps are constructed.

3.1 STS Content Maps

Most of the large, low resolution physical maps now publish are STS *content maps* (Green and Green 1991). A sequence-tagged site, or STS, is

- a unique sequence in the genome, along with
- a reliable biochemical assay for determining whether or not any given segment of DNA contains that sequence.

Hence, one can determine with low probability of error whether or not a clone contains any given STS. In this case, the "fingerprint" for a clone is the collection of STSs it contains.

Figure 1 contains a diagram of a toy example which we will use to describe issues in STS content mapping. Each horizontal line represents a clone. In the diagram, we show five clones, labeled 1 through 5. The five clones overlap in the way indicated, although we don't know that, of course. Each vertical arrow represents an STS. In the diagram, we show five STSs, labeled a through e. The task is to use information about which clones contain which STSs to determine the correct order of the STSs. If we can determine without error which clones contain which STSs, the following algorithm will produce a correct ordering.

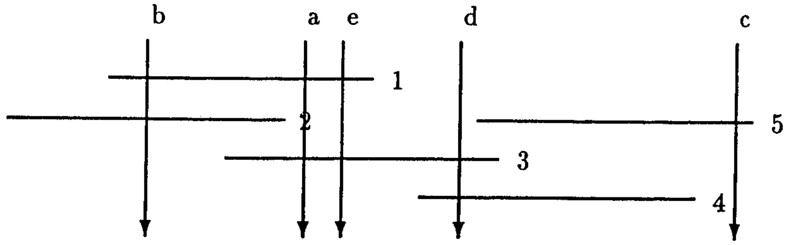


Figure 1: A simple example with five clones and five STSs.

First, construct an incidence matrix A containing one row per clone and one column per STS probe. Let $A_{ij} = 1$ whenever clone i contains probe j, and 0 otherwise. For our example in Figure 1, the matrix would be

		STSs				
		a	*b*	*c*	*d*	*e*
Clones	1	1	1	0	0	1
	2	0	1	0	0	0
	3	1	0	0	1	1
	4	0	0	0	1	0
	5	0	0	1	0	0

Each ordering of the probes corresponds to a permutation of the columns of A, which we can represent by AP, where P is a permutation matrix. As the clones in Figure 1 are intervals, and we have perfect detection, it is immediately clear that correct orderings P must permute the columns of A so that all the ones in each row of AP appear consecutively. Conversely, any permutation P for which AP has all ones in each row appearing consecutively corresponds to a correct probe ordering. Thus the problem reduces to finding all permutation matrices P for which AP has all the ones in each row appearing consecutively.

An incidence matrix whose columns can be permuted so that all the ones in its rows appear consecutively is said to have the *consecutive ones property for rows*. Fortunately, it is easy to check if a matrix has the consecutive ones property for rows. Booth and Lueker (1976) describe linear time algorithms which perform the check and return all correct permutations in a data structure called a "PQ tree". Hence, if the data are perfect, the problem is solvable in linear time.

In our example, the two permutations (b, a, e, d, c) and (b, e, a, d, c) both produce the identical permuted matrix:

		STSs				
		b	*a*	*e*	*d*	*c*
Clones	1	1	1	1	0	0
	2	1	0	0	0	0
	3	0	1	1	1	0
	4	0	0	0	1	0
	5	0	0	0	0	1

and hence both are orderings consistent with the data A. Also note that the locations of the runs of ones in the rows of AP provide an indication of precisely what spatial relationships among the clones can be deduced from the data.

Unfortunately, the data are *never* perfect. False negative rates of up to ten percent are not unusual (S. Lewis, private communication). Even more unfortunately, the existence of errors renders the problem much more difficult. The consecutive ones property is lost, and the problem of finding some nearby matrix A' which does have the consecutive ones property is, in general, NP hard.

Current approaches to handling data with errors treat the problem as one of combinatorial optimization. In general, combinatorial optimization problems involve searching over some large, but finite space in an attempt to minimize some objective function defined on elements of that space. Issues to be resolved include the structure of the space, the nature of the objective function, and the strategy used to search the space. In the case of STS content mapping, the search space is the space of all permutations on n letters, where n is the number of probes. The objective function is usually something like total number of runs of ones, or perhaps minus a pseudo log-likelihood of the data given the underlying probe order. At any given step in the search, the next permutation to be tested is determined heuristically, and simulated annealing is often used to escape from local minima.

LLNL has taken this approach in their attempts to produce an integrated map of chromosome 19. The data for LLNL's integrated map consists of over 2800 probes on 725 different segments of DNA. An even larger example is provided by Genethon's 1st Generation STS map (Cohen et al. 1993), which contains information about 2100 STSs and 6580 clones. Such large map construction efforts take many hours to compute, even when run on large workstations. Some recent work by Karp and his colleagues (R. Karp, private communication) indicates that solutions might be able to be more quickly computed by treating the problem as a Hamming distance Traveling Salesman Problem (TSP), and exploiting the wealth of heuristics developed to solve TSP problems.

3.2 Maps Based on "Random" Fingerprinting

STS content maps are not the only kind of physical map currently being constructed. One can also build maps of clone libraries "bottom-up" by a two stage process:

- use a fingerprint-based similarity measure to measure the similarity of any pair of clones in the library, and then
- use this similarity measure in a clustering procedure (Mardia, Kent, and Bibby 1979) to construct contigs.

The type of similarity measure used depends in large part on the nature of the fingerprint data. LLNL relied on a probability-based fingerprint when it used this approach as its first step in constructing a map of chromosome 19. In this situation, we obtain a random "match" vector D_{ij} for each pair of clones i and j. In addition, we have a simplified statistical model which enables us to compute $\Pr(D_{ij} \mid t)$, where $t \in [0,1]$ is the proportion of DNA shared by the two clones. Using this model, we can compute the posterior odds of overlap, given the data, up to a constant:

$$\frac{\Pr(\text{overlap} \mid D_{ij})}{\Pr(\text{no overlap} \mid D_{ij})} \propto \frac{\Pr(D_{ij} \mid \text{overlap})}{\Pr(D_{ij} \mid \text{no overlap})} = L(i,j)$$

where

$$(1) \qquad L(i,j) = \frac{\int_{t\in(0,1]} \Pr(D_{ij} \mid t)\, dP(t \mid t > 0)}{\Pr(D_{ij} \mid t = 0)}$$

We then use $\log L(i,j)$ as our similarity measure in a "smarter-than-average" single-linkage clustering procedure (T. Slezak, personal communication).

Now, computing $L(i,j)$ can be quite laborious. In our case, we have over 10^4 clones to assemble into contigs. Hence, we need to compute over 5×10^7 different $L(i,j)$ values to assemble a map, where each $L(i,j)$ is a numerical integration. Currently this process, even with several heuristic screening procedures to screen out obviously non-overlapping clones, runs several days on a network of over 30 workstations.

3.3 Restriction Maps for Validating Contigs

Once we have a putative map for a set of clones, we then need to validate the overlap configuration among the clones. We do so by constructing a *restriction map* of the clones. Constructing these maps rapidly currently poses a large, and as yet unsolved, computational challenge.

Figure 2 shows an example of a restriction map of a large contig containing twenty-eight clones, labeled F17252 through D716. (The labels are meaningful to the biologist, but are irrelevant to this discussion.) The DNA in each clone is represented by a horizontal line proportional to its length in bases. Each tick mark on each line represents a *restriction site*: a specific sequence (in this case GAATTC) that will recognized by a particular restriction enzyme. Under the right conditions, restriction enzyme molecules will bind to DNA molecules at restriction sites and cut the DNA into fragments whose sizes can be measured. For instance, the five tick marks on clone F6320 indicate that that clone contains five restriction sites, and that when digested, it will produce six fragments whose relative sizes are indicated by the distances between the tick marks. The line of tick marks at the bottom of the figure indicate the positions and distances between all of the sites in the stretch of DNA spanned by the contig.

One begins to construct a restriction map by digesting each clone and measuring the lengths of the resulting fragments. Then, given the list of clones and observed fragment sizes for each clone, one attempts to lay out the clones and line up all the fragments to produce a map like in Figure 2.

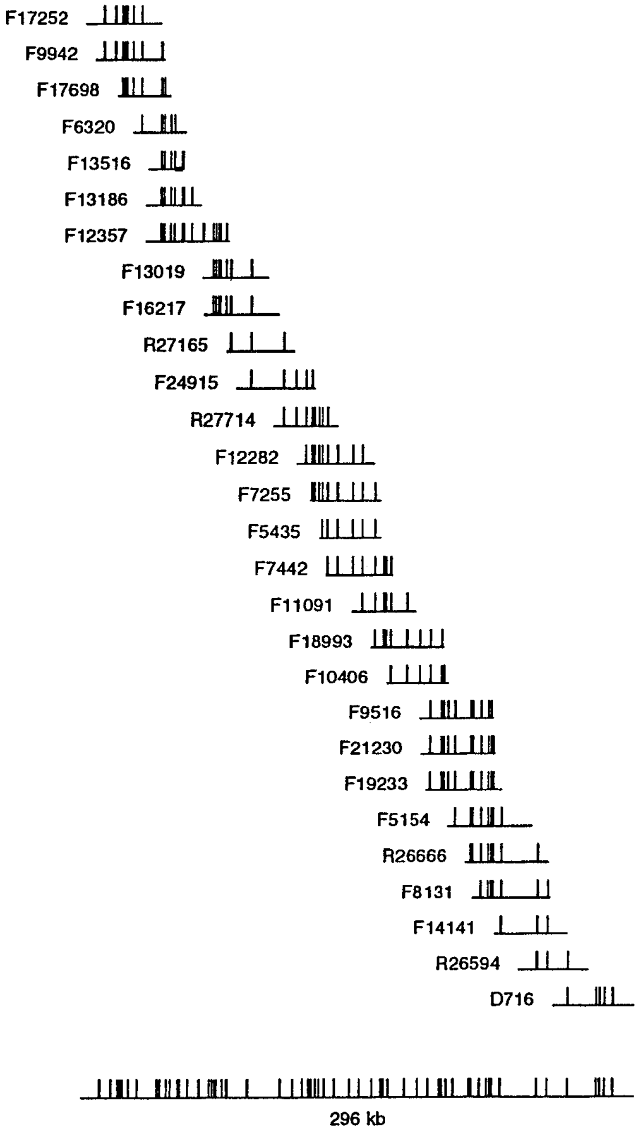


Figure 2: a restriction map of a large contig.

Currently, these maps are all produced manually, by an experienced mapper using a spreadsheet. *There are no automatic programs to produce these maps from a collection of clones and measured fragment sizes.* A number of issues complicate the construction of these maps. First, the problem is combinatorially explosive. Figure 3 shows a graph of the number of possible consistent, topologically distinct arrangements of clone beginnings and endings, as a function of the number of clones in a contig (Newberg 1993). Note the log scale.

Second, the measurement of fragment sizes is approximate and incomplete. The measurements are approximate in that, under good conditions, fragment lengths can be measured to within about one-half percent (Lamerdin and Carrano 1993). In addition, the measurements are incomplete in that it is sometimes difficult to determine exactly how many fragments of a given size have been digested. Also, there is left censoring: very small fragments are sometimes not measured at all.

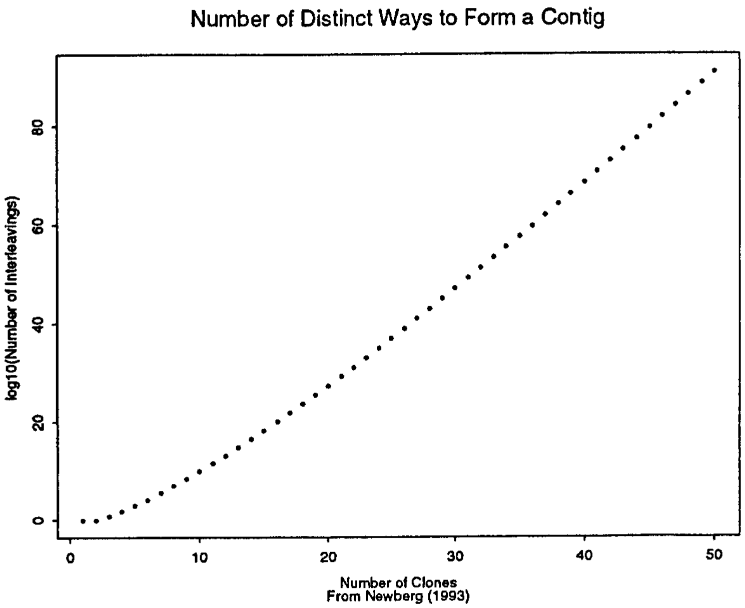


Figure 3: number of distinct interleavings versus contig size.

Third, perusal of the map will reveal that the sizes of the first and last fragments in a clone do not match the sizes of interior fragments of other clones. This is because the ends of the clones do not correspond to restriction sites for the enzyme used to create the restriction map. The sizes of the end fragments simply must not exceed the sizes of the interior fragments with which they are matched. Of course, the map constructor does not know beforehand which fragments are the end fragments.

Potential programs have another barrier to overcome: experienced mappers can assemble an "average" map in about an hour, based on good information about the approximate order of the clones. This apparent ability to recognize patterns in fragment sizes makes expert human mappers tough competitors to any program.

4 Getting It Right

The analysis and algorithms which go into a map assembly program can be quite complex. For instance, the integrand in Equation 1 involves several terms which incorporate assumptions about the data generation and error contamination processes, and must be numerically integrated to provide the numerator to the integrated likelihood ratio. To add to the computational burden, Equation 1 must be evaluated over 5×10^7 times during the construction of a map. To make the overall map construction process feasible, this computation must fully optimized. The problem we faced was how to ensure that the analysis we performed and algorithms we designed were fully specified and faithfully implemented.

Our solution to this problem, arrived at only after other, more ad hoc methods failed, was to describe what we wanted to do in the very high-level language implemented by Splus (Statistical Sciences, Inc. 1991). This specification could then be tested and debugged by executing it against sample data. After the specification was debugged, it was then reimplemented in C for speed of execution. After the C code was tested and debugged, the answers it produced for sample data could then be compared with the answers produced by the specification. Any differences represented bugs in the specification, implementation, or both.

This simple method of operational specification proved invaluable to us in a number of ways. First, it highlighted communication problems and definitional ambiguities between designer and implementor. Problems with defining exactly what a "match" meant, and how it was to be implemented, were quickly spotted and nailed down. Second, we found that as often as not, it was the specification that was ambiguous, indicating a need for further thought on the part of the designers. Third, having an executable specification could guide the debugging process, providing answers to partially complete calculations. Finally, the iteration process between designer and implementor converged quite rapidly, producing complex working software much more quickly than had been possible in the past. The technique was so successful that we now use it on *all* our software that has a significant mathematical or statistical component.

5 Summary

In this paper we have described several kinds of physical maps and outlined the methods currently employed for constructing these maps. These methods are characterized by being computationally intensive, combinatorially complex, and sometimes containing a considerable statistical component. It is clear from the descriptions that many computational and statistical challenges remain to be overcome.

One important challenge that must be addressed is how to parallelize the computational burden. For some tasks, this is easy: each of the 5×10^7 values of Equation 1 is an independent computation. Given a shared database and a way to communicate tasks to various workstations, parallelization becomes a matter of dispatching computations to free workstations and receiving the results. The Human Genome Center at LLNL has implemented such a scheme for its network of over thirty workstations. However, for many tasks, such as combinatorial optimization using simulated annealing, it is not clear how to parallelize the computation.

Another unsolved issue is how to combine information from various sources. The map of chromosome 19 integrates information on well over a dozen different types of probes and DNA regions, each with its own size, probe resolution, and error characteristics. At the present time, most of this data is treated democratically, ignoring the special features of each data type.

Finally, current procedures for constructing maps provide no information about the reliability of the resulting map. Developing statistics-based methods for map construction could provide a first step towards assessing the uncertainty of the resulting map as well as the sensitivity of the map to features in the underlying data.

References

Booth, K. S. and G. S. Lueker (1976). Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences 13*, 335–379.

Brown, T. A. (1990). *Gene cloning: an introduction* (2nd ed.). Chapman and Hall.

Cohen, D., A. Chumakov, and J. Weissenbach (1993). A first-generation physical map of the human genome. *Nature 366*, 698–701.

Green, E. D. and P. Green (1991). Sequence-tagged site (STS) content mapping of human chromosomes: theoretical considerations and early experiences. *PCR Methods and Applications 1*, 77—90.

Lamerdin, J. E. and A. V. Carrano (1993). Automated fluorescence-based restriction fragment analysis. *BioTechniques 15*, 294–300.

Mardia, K. V., J. T. Kent, and J. M. Bibby (1979). *Multivariate Analysis.* New York: Academic Press.

Nelson, D. O. and T. P. Speed (1994). Statistical issues in constructing high resolution physical maps. *Statistical Science* , in press.

Newberg, L. A. (1993). *Finding, evaluating, and counting DNA physical maps.* Ph. D. thesis, University of California, Berkeley.

Olson, M. V. (1993). The human genome project. *Proc. Natl. Acad. Sci. USA 90*, 4338–4344.

Statistical Sciences, Inc. (1991). *S-Plus User's Manual.*

A SIMULATION STUDY TO EVALUATE THE PERFORMANCE OF A NEW VARIABLE SELECTION METHOD IN REGRESSION

by

Gonzalo R. Mendieta†, Shahar Boneh‡, Roxy Walsh‡

†Universidad San Francisco de Quito
Quito, Ecuador
gonzalo@mail.usfq.edu.ec

‡Wichita State University,
Wichita, KS 67260-0033
boneh@twsuvm.uc.twsu.edu
walsh@twsuvm.uc.twsu.edu

ABSTRACT: The performance of new stepwise method for variable selection in regression will be evaluated in a large scale simulation study. This method is an extension of principal components regression. The on-going simulation is described. Preliminary results and previous tests on well known data sets show that the method is quite promising and may worked better that other methods in certain situations.

1. INTRODUCTION

Variable selection in regression is necessary when data are collected on a large number of variables, often correlated, while the goal is to obtain a model with only a few predictor variables. There are many variable selection methods commonly used, these include forward, backward and stepwise methods, or exhaustive search methods (using various criteria). While these methods often yield good outcomes, they have their shortcomings. Exhaustive search procedures may be very costly or even unfeasible in large scale problems, while systematic algorithms may sometimes fail to detect the best predictive subset of variables. For a comprehensive survey of variable selection methods, we refer to Miller [10].

Principal component regression, a well known and effective technique for reducing the dimensionality of the space of predictors, has the shortcoming that there is no corresponding reduction in the number of original variables. Jeffers [4] was the first to show that principal component analysis can be utilized to reduce the number of original variables. Realizing that the principal component transformation may be more informative than previously thought, more efforts were made in this direction in the subsequent years, most notably by Jolliffe ([5], [6], [7]), Hawkins [2], and Mansfield, Webster & Gunst [9].

Recently, a new method to select predictor variables based on principal components was proposed by Boneh & Mendieta [1]. The method is stepwise in nature, and it is based on repeated selections of principal components and inversions to the original variables. The main idea of this method is to combine the advantages of stepwise selection with those of principal component regressions. The method was tested on several benchmark data sets, and produced good results. (An example is given in Boneh & Mendieta [1]). Having established that the new method is statistically sound, is was called upon to further study its performance. In particular, to identify its strengths and possible weaknesses, and to determine in what circumstances it may be preferable to other methods.

The goal of this paper is therefore to give a brief introduction to the method and to report on the design of an on-going simulation study aim at answering the above questions. In Section 2 we briefly describe the selection method, and in Section 3 we describe the layout and goals of the simulation. Some general remarks are given in Section 4.

2. THE SELECTION METHOD

We consider the standard linear regression model $Y = X\beta + \mathcal{E}$, where Y is an $n \times 1$ vector of responses, $X = [X_1,...,X_p]$ is an $n \times p$ full rank matrix of predictor variables, β is a $p \times 1$ vector of unknown parameters, and $\mathcal{E}$ is an $n \times 1$ vector of uncorrelated and normally distributed random errors with mean 0 and common variance σ^2. Without loss of generality, all the variables are assumed to be standardized (with mean 0 and variance 1). Thus $[X^T X; X^T Y]$ is the sample correlation matrix.

We assume that the reader is familiar with the basic concepts of principal component analysis. Otherwise, as good references on the subject we recommend Jolliffe [8] or Jackson [3].

Prior to starting the selection, we select a fixed level α through out the process.

Step 0: Selection of the first variable

0.1. Obtain the principal components, $W = [W_1, ..., W_p]$, of $[X_1, ..., X_p]$.

0.2. Fit the model $Y=W\gamma + \mathcal{E}$, and let $W_{(s)}$ be the subset of W containing the principal components for which the regression coefficient $\hat{\gamma}_j$ is significant at level α.

0.3. If $W_{(s)}$ is empty, the selection process terminates with the conclusion that no predictor variables should be included in the model. Otherwise, let SSE_j, j=1,...,p, denote the error sum of squares when X_j is regressed on $\mathbf{W}_{(s)}$. The first predictor selected is the one for which SSE_j is minimal.

Continue to select additional variables according to the following general steps:

Step I: Let $Y_{(s)}$ and $X_{(r)}$ be respectively the sets of the previously selected variables and the remaining unselected variables. Regress each variable in $X_{(r)}$ on all the variables in $X_{(s)}$ and obtain the corresponding vectors of standardized residuals $\{E_j\,,\ j \in (r)\}$.

Step II: Obtain the principal components W, of $\{E_j\}$, and regress Y on $X_{(s)}$ and W.

Step III: Let $W_{(s)}$ denote the subset of W containing the principal components with significant regression coefficients (at level α).

Step IV: If $W_{(s)}$ is empty, the selection process terminates. Otherwise, let SSE_j, $j \in (r)$, denote the error sum of squares when E_j is regressed on $W_{(s)}$. The next variable selected is the one corresponding to the minimal SSE_j.

After the selection of each variable, the previously selected variables are verified (essentially reversing the selection steps) as follows:

Step V: Let X_k denote the most recently selected variable, i.e., the one which was selected in the current step, and let $X_{(c)}$ denote the set of the previously selected variables. Regress each of the variables in $X_{(c)}$ on X_k and obtain the standardized residuals $E_{(c)}$.

Step VI: Obtain the principal components $W_{(c)}$ of $E_{(c)}$, and regress Y on X_k and $W_{(c)}$.

Step VII: If all the regression coefficients of $W_{(c)}$ are significant at level α, we conclude that all the variables in $X_{(c)}$ should stay in the model.

Step VIII: Otherwise, one variable from $X_{(c)}$ must be dropped. To determine which one, let $W_{(n)}$ be the subset of $W_{(c)}$ containing the principal components with the non-significant coefficients. Regress each residual vector in $E_{(c)}$ on $W_{(n)}$, and obtain SSE_j, $j \in (c)$. The variable in $X_{(c)}$ corresponding to the minimal SSE_j is dropped from the model.

The verification is then carried out again to check if additional variables in $X_{(c)}$ should be dropped. A predictor variable that was dropped is excluded from the pool of potential variables in all the future steps to avoid possible cycling in the process.

The process terminates when no principal components have significant regression coefficients in the selection step, or when the pool of predictor variables is depleted.

The method is described in detail in Boneh & Mendieta [1]. The following are the main formulas used in the implementation of the above steps. Proofs are given in [1].

(1) To select principal components in step 0.2 and the general steps III & VII, the hypothesis H_0: $\gamma_j = 0$ is tested by the t-test as follows:
Reject H_0 if $\sqrt{\frac{n-p-1}{\lambda_j}} \cdot \frac{|\hat{\gamma}_j|}{\sqrt{SSE}} > t_{n-p-1,\alpha/2}$, where, $\hat{\gamma}_j = \frac{1}{\lambda_j} V_j^T E^T Y$ and $SSE = 1 - (Y^T E) V \Lambda^{-1} V^T (E^T Y) - (Y^T X_{(s)})(X_{(s)}^T X_{(s)})^{-1}(X_{(s)}^T Y)$. Here E denotes the matrix of standardized residuals of the regression of the remaining unselected variables on $X_{(s)}$. When selecting the first variable, E is replaced by X and $X_{(s)}^T$ is empty. Note that the matrices V and Λ are computed each time from a different set of variables.

(2) SSE_j, $j \in (r)$ (Step IV), is calculated by $SSE_j = \sum_{k \notin (s)} v_{jk}^2 \lambda_k$.

(3) Let E_j be the vector of standardized residuals when regressing X_j ($j \in (r)$) on $Y_{(s)}$ (Step I). Denote by $E = \{E_j,\ j \in (r)\}$. All we need for the next selection is $E^T E$ and $E^T Y$, which are given by $E^T E = \left(\frac{A_{ij}}{\sqrt{A_{ii}\,A_{jj}}}\right)$, $i, j = 2, \ldots, p$, and $E^T Y = \left(\frac{B_j}{\sqrt{A_{jj}}}\right)$, $j = 2, \ldots, p$,
where, $A_{ij} = (X_i^T X_j) - (X_i^T X_{(s)})(X_{(s)}^T X_{(s)})^{-1}(X_{(s)}^T X_j)$ and $B_j = (Y^T X_j) - (Y^T X_{(s)})(X_{(s)}^T X_{(s)})^{-1}(X_{(s)}^T X_j)$.

An important feature that emerges from Formulas (1)-(3) is that the method can be carried out with the correlation matrix only, without direct use of the raw data. This feature enhances the computational efficiency and convenience of the algorithm.

3. LAY OUT OF THE SIMULATION EXPEDIMENT

In this section we describe the layout of our simulation study. In designing our simulation experiment we made used of some results regarding the design of simulation experiments in regression given in [11].

Generation of the data:

All data sets to be considered in this study will be generated from a normal distribution with mean 0 and covariance given by the $(p+1) \times (p+1)$ matrix

$$C = \begin{pmatrix} \rho_X & \rho_{XY} \\ \rho_{XY} & 1 \end{pmatrix}.$$

Several types of correlations ρ_X between the predictors will be considered. The correlations ρ_{XY} between the response and the predictors are such that they correspond to particular specifications of the slopes in the model $Y = X\beta + \mathcal{E}$.

Factors to be considered:

Our simulation layout corresponds to a factorial experiment with the following factors:

1. *Number of predictors in the model:* We will consider models with 4 and 8 predictors.
2. *Number of predictors with non-zero slope:* We will consider models with 1 and 2 predictors with non-zero slopes.
3. *Sample size:* 50 and 100 data points.
4. *Type of correlation structure between the predictors:* We will study correlations of the form $\rho_X = \begin{pmatrix} A_q & 0 \\ 0 & I_{p-q} \end{pmatrix}$ where, I is the identity matrix of order $p - q$ and A is one of the following matrices:

   Equi-correlation,

$$A = \begin{pmatrix} 1 & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho \\ \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & 1 \end{pmatrix}.$$

   Markovian,

$$A = \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{pmatrix}.$$

   Equi-predict,

$$A = \begin{pmatrix} 1 & \rho_1 & \rho_2 & \rho_3 \\ \rho_1 & 1 & \rho_1 & \rho_2 \\ \rho_2 & \rho_1 & 1 & \rho_1 \\ \rho_3 & \rho_2 & \rho_1 & 1 \end{pmatrix}.$$

   Partitioned,

$$A = \begin{pmatrix} 1 & \rho_1 & 0 & 0 \\ \rho_1 & 1 & 0 & 0 \\ 0 & 0 & 1 & \rho_2 \\ 0 & 0 & \rho_2 & 1 \end{pmatrix}.$$

   The predictors associated with I will be used as *noise* variables.

5. *Values of the parameters*: We will set σ=1 and the values of the non-zero slopes will be selected in such a way that the 0.10 t-test for testing the hypothesis that β=0, in the model $Y = X_j\beta + \mathcal{E}$ has an approximate power of .90 and .99. The corresponding values of .43 and .62 for a sample size of 50, and .31 and .45 for a sample size of 100 can be obtained from results reported in [11]

For each of these factor-level combinations a total of 1000 different data sets will be generated. Both, our algorithm and the standard stepwise algorithm as implemented in S-plus will be run and analyzed.

4. MEASURES OF PERFORMANCE

The performance of the algorithms will be evaluated using the following measures:

1. *Mean Square Error of Prediction*: For each model a prediction data set consisting of 100 data points from the true model will be generated. The quantity

$$MSP = \tfrac{1}{100}\sum(\widehat{Y}_j - \mu_j)^2$$

   will be computed. Here $\widehat{Y}_j$ is the predicted response computed from the selected model, and μ_j is the true response at the j-th observation.
2. *Proportion of times the correct model is selected.*
3. *Proportion of time each of the predictors with non-zero slope was included in the final model.*
4. *The mean number of noise variables selected in the final model.*

In addition such numerical performance measures as speed and number of iteration will also be measured.

REFERENCES

[1] Boneh, S. & Mendieta, G.R. (1994) - Variable selection in regression models using principal components. *Comm. Stat. -Theory Meth.* **23,** 197-213.

[2] Hawkins, D.M. (1973) - On the investigation of alternative regression by principal component analysis. *Appl. Statist.* **22,** 275-286.

[3] Jackson, J.E. (1991) - *A User's Guide To Principal Components.* Wiley, New York.

[4] Jeffers, J.N. (1965) - Correspondence. *Statistician,* **15,** 207-208.

[5] Jolliffe, I.T. (1972) - Discarding variables in principal component analysis. I. Artificial data. *App. Statist.,* **22,** 21-31.

[6] Jolliffe, I.T. (1973) - Discarding variables in principal components analysis II. Real data. *Appl. Statist.,* **22,** 21-31.

[7] Jolliffe, I.T. (1972) - A nore on the use of principal components in regression. *Appl. Statist.,* **31,** *300-303.*

[8] Jolliffe, I.T. (1972) -*Principal Component Analysis.* Springer-Verlag, New York.

[9] Mansfield, E. R., Webster, J.T. & Gunst, R.F. (1977) - An analytic variable selection technique for principal component regression. *Appl. Statist.,* **36,** 34-40.

[10] Miller, A. J. (1990) -*Subset Selection in Regression.* Chapman and Hall, London.

[11] Thall, P.F., Simon, R. & Grier, D.A. (1992) - Test-based variable selection via cross-validation. *J. of Comp. and Graph. Statistics,* **1,** 41-61.

Saddlepoint Approximations for Robust M regression

Edgar Acuna, University of Puerto Rico-Mayaguez Campus
Department of Mathematics, Mayaguez, PR 00681

Abstract

In this paper, first we use saddlepoint methods to approximate the density of an estimator T_n for a p-dimensional parameter θ, that is given implicitly as the solution of p nonlinear equations $\sum_{i=1}^{n} \psi_j(x_i, T) = 0, \quad j = 1, \cdots, p$. Here, the X_i are i.i.d. random variables with density $f(x, \theta)$ and ψ_j is a nondecreasing function satisfying certain mild regularity conditions. The one-dimensional case was treated by H. Daniels (Biometrika, 1983).

Then, we find saddlepoint approximations for the densities of the least squares and the robust M estimator of regression $\mathbf{B_M}$. For the linear regression model $y_i = x_i^T\beta + e_i$, B_M satisfies the system of equations $\sum_{i=1}^{n} x_i\psi(y_i - x_i^T\beta) = \mathbf{0}$ where x_i^T is a p-dimensional row vector and β is a p-dimensional column vector. If $\psi(u) = u$ the least squares is obtained.

1 Introduction

Let $X_1, X_2, \cdots, X_n$ be i.i.d. random variables with density function $f(x)$, and generating moment function, $M(t) = \int_{-\infty}^{\infty} e^{tx} f(x)dx$ which converges for each real t in the interval (c_1, c_2) that contains zero.

Let $f_n(\overline{\mathbf{x}})$ the density function of the sample mean $\overline{\mathbf{x}} = \sum_{i=1}^{n} X_i/n$. H. Daniels (1954), applied saddlepoint methods of asymptotic analysis to find $f_n(\overline{\mathbf{x}}) = g_n(\overline{\mathbf{x}})[1 + O(\frac{1}{n})]$ where,

$$g_n(\overline{\mathbf{x}}) = \{\frac{n}{2\pi K''(T_0)}\}^{1/2} \exp(n(K(T_0) - T_0\overline{\mathbf{x}})) \qquad (1.1)$$

is called the saddlepoint approximation to the density $f_n(\overline{x})$. Here $K(T) = \log M(T)$ is the cumulant generating function, and T_0 is the saddlepoint, i.e $K'(T_0) = \overline{\mathbf{x}}$. Since $g_n(\overline{\mathbf{x}})$ does not integrate to 1, sometimes a renormalized saddlepoint approximation is used.

The Saddlepoint approximation improves the one given by the two-term Edgeworth expansion for $f_n(\overline{\mathbf{x}})$, which can give negative values for $\overline{\mathbf{x}}$ values far away from the mean μ.

The Saddlepoint approximation can also be obtained by using a conjugate family of densities for $f(x)$ defined by $f(x, \lambda) = \exp(\lambda x - K(\lambda))f(x)$, which has $\mu_\lambda = K'(\lambda)$ and variance $\sigma_\lambda^2 = K''(\lambda)$. Notice that

$$f_n(\overline{\mathbf{x}}) = f(\overline{\mathbf{x}}, \lambda) \exp(n[K(\lambda) - \lambda\overline{\mathbf{x}}]) \qquad (1.2)$$

Then, using an Edgeworth expansion for $f(\overline{\mathbf{x}}, \lambda)$ at its center and letting $\lambda = T_0$, we obtain the saddlepoint approximation (1.1). This procedure is called Tilted or indirect Edgeworth.

O. Barndorff-Nielsen and D. R. Cox (1979), extended Daniel's result to multivariate densities. Let $X_1, \cdots, X_n$ be p-dimensional random vectors with cumulant generating function $K(T)$ where T is in R^p. Then, the saddlepoint approximation to the density function $f_n(\overline{\mathbf{x}})$ of the p-dimensional mean $\overline{\mathbf{x}}$ is

$$g_n(\overline{\mathbf{x}}) = \{\frac{n}{2\pi}\}^{p/2}\{\frac{1}{|K''(T_0)|}\}^{1/2} e^{n(K(T_0) - T_0'\overline{\mathbf{x}})} \qquad (1.3)$$

where T_0 is the p-dimensional saddlepoint, T_0' its transpose and $|K''(T_0)|$ is the determinant of the matrix $K''(T_0) = [\frac{\partial K(T)}{\partial T_i \partial T_j}]$.

2 Saddlepoint methods for M-estimators

Let X be a random variable with density function $f(x,\theta)$, where θ is an unknown parameter. An M-estimator $\hat{\theta}_n$ of θ based in a random sample $X_1,\cdots,X_n$ is obtained by solving with respect to t

$$\sum_{i=1}^{n}\psi(x_i,t)=0 \qquad (2.1)$$

where ψ is a nondecreasing function.

Notice that for $\psi(x,t)=x-t$ we obtain the sample mean $\mathbf{\bar{x}}$, and for $\psi(x,t) = \partial f(x,t)/\partial t$ we obtain the ML estimator of θ. If $\psi(u)$ is bounded then the estimator is said to be Robust, for instance for $\psi(u)=\min(k,\max(u,-k))$ we obtain the Huber estimator.

H. Daniels (1983) found the saddlepoint approximation to the density of $f_n(\tilde{\theta}_n)$, where θ_n solves the equation (2.1). At the point $\theta_n = a$, f_n is approxinated by

$$g_n(a) = \{\frac{n}{2\pi K''(T_0,a)}\}^{1/2}[\frac{-K^*(T_0,a)}{T_0}e^{nK(T_0,a)}] \qquad (2.2)$$

where $K(T,a)$ is the cumulant generating function of $W = \psi(X,a)$, here a is a fixed value, $K'(T_0,a)=0$, and K* represents the derivative of K with respect to a.

Now let us treat the multiparametric case. Here θ is in R^p and the M-estimator $\hat{\theta}_n$ is the solution (in t) of the system of p nonlinear equations

$$\sum_{i=1}^{n}\psi_j(x_i,t)=0, \quad j=1,\cdots,p \qquad (2.3)$$

Let us consider the random vector $\psi(x,\mathbf{a}) = (\psi_1,\cdots,\psi_p)$. From now on consider the nonegative integral vector $v=(v_1,\cdots,v_p)$. Further write $|v| = \sum_{i=1}^{p} v_i$, $v! = v_1!\cdots v_p!$ and $D^v = (D_1)^{v_1}\cdots(D_p)^{v_p}$ for the v-th derivative with respect to θ .

If the following conditions are satisfied (see Huber (1981) pg. 132):

A1. $E_{\mathbf{a}}[\psi(x,\mathbf{a})]=0$

A2. $E_{\mathbf{a}}[||\psi(x,\mathbf{a})||^2] < \infty$ and there exists an $\epsilon > 0$ such that

$$E_{\mathbf{a}}[\max_{||\theta-\mathbf{a}||<\epsilon} ||D\psi(x,\theta)||^2] < \infty$$

A3. The matrices $A = (E_{\mathbf{a}}D_r\psi_j(x,\mathbf{a}))_{1\le r,j\le p}$ and $C = \mathbf{COV}[\psi(x,\mathbf{a})] = E_{\mathbf{a}}[\psi_i(x,\mathbf{a})\psi_j(x,\mathbf{a})]$ are nonsingular.

Then $T_n = \sqrt{n}(\hat{\theta}_n - \mathbf{a})$ has a limiting p-variate normal distribution with mean 0 and dispersion matrix $A^{-1}C(A^{-1})^T$.

Replacing the condition **A2** by

A2'. $E_{\mathbf{a}}[||D^v\psi(x,\mathbf{a})||^3] < \infty$ for $|v| = 1,2$ and there exists an $\epsilon > 0$ such that

$$E_{\mathbf{a}}[\max_{||\theta-\mathbf{a}||<\epsilon} ||D^v\psi(x,\theta)||^3] < \infty$$

if $|v| = 3$ for $j = 1,\cdots,p$.

The two term Edgeworth expansion for the distribution function of $\sqrt{n}(\hat{\theta}_n - \mathbf{a})$ can be obtained from the theorem 3 of Bhattacharya and Ghosh (1978, page 440). Thus

$$P(\sqrt{n}(\hat{\theta}_n - \mathbf{a}) \in B) = \int_B [1+n^{-1/2}P_1(x)]\phi_M(x)dx + o(n^{-1/2}) \qquad (2.4)$$

uniformly in B that belongs to the Borel system of R^p. Here ϕ_M stands for the p-variate normal density with mean 0 and covariance matrix $M = A^{-1}C(A^{-1})^T$. Also $P_1(x)$ is a polynomial not depending on n whose coefficients are themselves polynomials on the moments of order 3 or less of $\psi(x,\mathbf{a})$.

Let us consider the conjugate family of densities for $f(x)$ given by

$$f(x,\lambda) = e^{\lambda'\psi - K_\psi(\lambda,\mathbf{a})}f(x)$$

where $K_\psi(\lambda,\mathbf{a})$ is the cumulant generating function of the random vector $\psi(x,\mathbf{a})$. Notice that $E_\lambda[\psi(x,\mathbf{a})] = K'_\psi(\lambda,\mathbf{a})$ and $\mathbf{COV}_\lambda[\psi(x,\mathbf{a})] = K''_\psi(\lambda,\mathbf{a})$. It is easy to prove that

$$f_{\hat{\theta}_n}(\mathbf{a}) = e^{nK_\psi(\lambda,\mathbf{a})}f_{\hat{\theta}_n}(\mathbf{a},\lambda) \qquad (2.5)$$

Notice that $f_{\hat{\theta}}(w,\lambda) = n^{p/2}f_{T_n}(\sqrt{n}(w-\mathbf{a}),\lambda)$. Therefore $f_{\hat{\theta}_n}(\mathbf{a},\lambda) = n^{p/2}f_{T_n}(\mathbf{0},\lambda)$. Choosing

$\lambda = T_0$ such that $K'_\psi(T_0, \mathbf{a}) = 0$, it follows from 2.4 that

$$f_{T_n}(\mathbf{0}, \lambda) = \frac{|A|}{(2\pi)^{p/2}|C|^{1/2}} \qquad (2.6)$$

where $|A|$ is the determinant of the matrix A which is computed under the conjugate distribution, and $C = K''_\psi(T_0, \mathbf{a})$ where T_0 denotes the p-dimensional saddlepoint. Finally turns out that the saddlepoint approximation to the density function of $\hat{\theta}_n$ at the point $\mathbf{a}$ is

$$g_n(\mathbf{a}) = \{\frac{n}{2\pi}\}^{p/2}|K''(T_0, \mathbf{a})|^{-1/2}|A|e^{nK(T_0,\mathbf{a})} \qquad (2.7)$$

Usually the saddlepoint has to be computed numerically over a grid of values $\mathbf{a}$. An equivalent result to (2.7) has been obtained by Field (1982), who also shows some numerical examples.

Example 1. Location and scale estimation in a normal density

Let us consider a Normal randon variable X with mean μ and standard deviation σ, both of them unknown. Let $\mathbf{a} = (a, b)$ where a and b are the least squares estimators of μ and σ respectively. In this case $\psi_1(x, \mathbf{a}) = \frac{(x-a)}{b}$ and $\psi_2(x, \mathbf{a}) = \frac{(x-a)^2}{b^2} - 1$.

After long computations we obtain

$$K(T, \mathbf{a}) = -t_2 + \frac{(\theta - a)^2}{b^2}t_2 + \frac{(\theta - a)t_1}{b}$$

$$+\frac{\sigma^2(2(\theta - a)t_2 + bt_1)^2}{2b^2(b^2 - 2\sigma^2 t_2)} + \frac{1}{2}log(\frac{b^2}{b^2 - 2\sigma^2 t_2})$$

The saddlepoint is $T_0 = (-\frac{(\theta-a)b}{\sigma^2}, \frac{b^2-\sigma^2}{2\sigma^2})$. Also

$$K(T_0, \mathbf{a}) = \frac{1}{2} - \frac{b^2}{2\sigma^2} - \frac{(\theta - a)^2}{2\sigma^2} + \log(\frac{b}{\sigma})$$

$|K''(T_0, \mathbf{a})| = 2$ and $|A| = \frac{2}{b^2}$. Then, the saddlepoint approximation is given by

$$g_n(\mathbf{a}) = \frac{n}{\pi\sqrt{2}b^2}\{\frac{b^2}{\sigma^2}\}^{n/2}e^{-\frac{n(\theta-a)^2}{2\sigma^2}+\frac{n}{2}-\frac{nb^2}{2\sigma^2}}$$

which results to be exact except for the constant.

3 Saddlepoint approximations for estimators of regression

Let us consider the multiple regression model

$$y_i = x_i^T\beta + e_i \qquad i = 1, \cdots, n \qquad (3.1)$$

where $e_1, \cdots, e_n$ are i.i.d random variables with common distribution F; $x_1^T, \cdots, x_n^T$ are known nonrandom p-dimensional row vectors and β is the $p \times 1$ vector of unknown parameters. We will use also the following notation:

$X = (x_1^T, \cdots, x_n^T)$ represents a design matrix and X' its tranpose. Notice that $X'X = \sum_{i=1}^n x_i x_i^T$.

$H = X(X'X)^{-1}X'$ is a projection matrix with diagonal element h_{ii}.

Next we will discuss the saddlepoint approximation for the density of the least squares estimator of regression, which is based in the fact that can be expressed as a linear combination of the $e_i's$. Later we will treat the case of the M-estimator of regression.

3.1 Saddlepoint approximation in least squares regression

The least squares estimator $\hat{\beta}$ of β in the regression model (3.1) is given by $\hat{\beta} = (X'X)^{-1}X'Y$.

Huber (1981, pg. 159) proved that under the following conditions

B1. $e_i's$ are i.i.d with mean 0 and finite variance σ^2.

B2. X has full rank p.

B3. $max_{1\leq i\leq p}h_{ii} \to 0$

Then $T_n = (X'X)^{1/2}(\hat{\beta} - \beta)$ has a limiting p-variate normal distribution with mean 0 and dispersion matrix $\sigma^2 I_p$, where I_p is the identity matrix of order p.

Under the conditions **B1**, **B2** and the ones given below

B3'. $e_i's$ have finite s-th absolute moment, for some integer $s \geq 3$ and, $\overline{\lim}\frac{1}{n}\sum_{i=1}^n ||x_i||^s \leq \infty$.

B4'. $\underline{\lim}\lambda_n/n > 0$ and $M_n = O(n^\delta)$ for some $\delta \in [0, 1/2)$. Here λ_n =the smallest eigenvalue of X'X and $M_n = \max_{1\leq i\leq n}||x_i||$.

Qumsiyeh (1990) obtained the following two-term Edgeworth expansion for the density function q_n of T_n

$$(1+||x||^4)|q_n(x)-[1+n^{-1/2}P_1(-D,\{\overline{\chi}_v\})]\phi_{\sigma^2}(x)| = O(n^{-1}) \qquad (3.2)$$

uniformly in $x \in R^p$. Here ϕ_{σ^2} represents the p-variate normal density with mean 0 and covariance matrix $\sigma^2 I_p$. Also $P_1(-D,\{\overline{\chi}_v\})\phi_{\sigma^2} = -\sum_{|v|=3}\frac{\overline{\chi}_v}{v!}D^v\phi_{\sigma^2}(x)$ and, $\overline{\chi}_v = \frac{1}{n}\sum_{i=1}^n \chi_v(Z_i)$ where $\chi_v(Z_i)$ denotes the v-th cumulant of $Z_i = n^{1/2}(X'X)^{-1/2}x_ie_i$, which are independent with mean 0.

Now let us derive the saddlepoint apprximation to the density of T_n. Let $T_n = \sum_{i=1}^n d_ie_i$ being $d_i = (X'X)^{-1/2}x_i$ a $p \times 1$ vector. Then $K_{T_n}(\mathbf{t}) = \sum_{i=1}^n K_{e_1}(d_i'\mathbf{t})$, where $K_{e_1}(\cdot)$ stands for the cumulant generating function of the error e_1. Also $K_{T_n}'(\mathbf{t}) = \sum_{i=1}^n d_iK_{e_1}'(d_i'\mathbf{t})$ and $K_{T_n}''(\mathbf{t}) = \sum_{i=1}^n d_iK_{e_1}''(d_i'\mathbf{t})d_i'$. On the other hand

$$f_{T_n}(\mathbf{a}) = e^{K_{T_n}(\lambda)-\lambda\mathbf{a}}f_{T_n}(\mathbf{a},\lambda) \qquad (3.3)$$

Notice that $E_\lambda[T_n] = K_{T_n}'(\lambda)$. Also $\mathbf{COV}[T_n] = K_{T_n}''(\lambda)$.

Choosing $\lambda = \mathbf{t_0}$ such that $K_{T_n}'(\mathbf{t_o}) = \mathbf{a}$ then from (3.2) and (3.3) the saddlepoint approximation to the density of T_n at the point $\mathbf{a}$ is as follows

$$g_n(\mathbf{a}) = \frac{e^{K_{T_n}(t_o)-t_o'\mathbf{a}}}{(2\pi)^{p/2}|K_{T_n}''(t_o)|^{1/2}} \qquad (3.4)$$

Example 2 Normally distributed errors

In this case $K_{e_1}(t) = \frac{t^2}{2}\sigma^2$, $K_{e_1}'(t) = t\sigma^2$ and $K_{e_1}''(t) = \sigma^2$. Since $\sum_{i=1}^n d_id_i' = I$, then $\mathbf{t_o} = \frac{\mathbf{a}}{\sigma^2}$ also $K_{T_n}''(\mathbf{t_o}) = \sigma^2 I_p$ and $K_{T_n}(\mathbf{t_o}) - \mathbf{t_o'a} = -\frac{\mathbf{a'a}}{2\sigma^2}$ yielding the saddlepoint approximation

$$g_n(\mathbf{a}) = \frac{1}{(2\pi\sigma)^{p/2}}e^{-\frac{\mathbf{a'a}}{2\sigma^2}}$$

which results to be exact.

3.2 Saddlepoint approximation for M regression

Let ψ a nondecreasing and bounded real-valued function, then an M-estimator B_M of β corresponding to ψ is defined as the solution (in t) of the vector equation

$$\sum_{i=1}^n x_i\psi(y_i - x_i^Tt) = \mathbf{0}$$

It is well known (Huber, 1981, pg. 165) that under the following conditions on the error distribution F, ψ and the design matrix X:

C1. ψ is twice differentiable and the second derivative ψ'' satisfies a Lipschitz condition of order ω for some $0 < 2\omega \leq 1$.

C2. $E(\psi(e_1)) \doteq 0$, and $\tau^2 = \frac{E\psi^2(e_1)}{E\psi'(e_1)^2} \in (0,\infty)$

C3. $\sum_{i=1}^n x_ix_i'$ is invertible for some $n \geq p$.

Then $T_n = (\sum_{i=1}^n x_ix_i^T)^{1/2}(B_M - \beta)$ has a limiting p-variate normal distribution with mean 0 and dispersion matrix $\tau^2 I_p$, where I_p denotes the identity matrix of order p.

Write $q = p(p+1)/2$ and for each $d_i = (d_{i1},\cdots,d_{ip})^T$ define the $q \times 1$ vector $c_i^T = (d_{i1}^2, d_{i1}d_{i2},\cdots,d_{i1}d_{ip}; d_{i2}^2, d_{i2}d_{i3},\cdots,d_{i2}d_{ip};\cdots;d_{ip}^2)$.

The spectral decomposition of the real symmetric matrix $\sum_{i=1}^n c_ic_i^T$ yields a $q\times q$ nonsigular matrix B of rank r such that

$$B(\sum_{i=1}^n c_ic_i^T)B' = \begin{vmatrix} I_r & 0 \\ 0 & 0 \end{vmatrix}$$

Let $B' = [B_1|B_2]$ where B_1 is of order $r \times q$. Define the column vector b_i by $b_i = B_1c_i$ for $1 \leq i \leq n$.

Let $\gamma_n = (\sum_i ||d_i||^6)^{1/4} + (\sum_i ||b_i||^4)^{1/2}$.

For $\delta > 0$, define $A_n(\delta) = \{i : 1 \leq i \leq n, (d_i't_1)^2 + (b_i't_2)^2 > \delta\gamma_n^2$ for all $t_1 \in R^p$ and $t_2 \in R^r$ with $||t_1||^2 + ||t_2||^2 = 1\}$.

Consider the following two additional condition:

C4. $\gamma_n = o(1)$.

C5. There exists $\delta > 0$ such that $\frac{-\log\gamma_n}{K_n(\delta)} = o(1)$ where $K(\delta) = \#[A(\delta)]$.

Under conditions **C1-C5**, Lahiri (1992b) has found the following two-term Edgeworth expansion for the distribution fumction of T_n

$$P(T_n \in B) = \int_B (1 + P_1(F,x))\phi_\tau(x)dx + o(\gamma_n) \tag{3.5}$$

uniformly in B that belongs to the Borel system of R^p. Here ϕ_τ stands for the p-variate variate normal with mean 0 and covariance matrix $\tau^2 I_p$ $P_1(F,x)$ is a polynomial, whose coeeficients are continuous functions of the finite moments of $\psi(e_1)$, $\psi'(e_1)$ and $\psi''(e_1)$.

Now let us obtain an approximated saddlepoint approximation for the density function of T_n.

Using equation (3.12) from Lahiri's paper (1992b, pg. 1560) we can write

$$T_n = \alpha^{-1}\sum_{i=1}^{n} d_i\psi(e_i) + R_{1n} \tag{3.6}$$

where $\alpha = E[\psi'(e_i)]$ and R_{1n} is a remainder.

Using (3.6) we can approximate the cumulant generating function of T_n as is suggested for Easton and Ronchetti (1986). Thus $K_{T_n}(\mathbf{t}) \approx \sum_{i=1}^{n} K_{\psi(e_1)}(\alpha^{-1}d_i'\mathbf{t})$, where $K_{\psi(e_1)}(\cdot)$ stands for the cumulant generating function of $\psi(e_1)$.

Also $K'_{T_n}(\mathbf{t}) \approx \sum_{i=1}^{n} \alpha^{-1}d_i K'_{\psi(e_1)}(\alpha^{-1}d_i'\mathbf{t})$ and $K''_{T_n}(\mathbf{t}) \approx \sum_{i=1}^{n} \alpha^{-2}d_i K''_{e_1}(\alpha^{-1}d_i'\mathbf{t})d_i'$.

Evaluating the above expressions at the saddlepoint $\mathbf{t_o}$ and replacing them in (3.4) we obtain an approximated saddlepoint approximation for the density of T_n.

References

[1] Barndorff-Nielsen, O. and Cox, D.R. (1979). Edgeworth and Saddlepoint approximations with statistical applications. *J. Roy. Statist. Soc. Ser B* **41**, 279-312.

[2] Bhattacharya, R.N. and Ghosh, J.K. (1978). On the validity of the formal Edgeworth expansion. *Ann. Statist.* **6**, 434-451.

[3] Bhattacharya, R.N. and Ranga Rao, R. (1986). *Normal Approximations and Asymptotic expansions.* Krieger, Malabar, Florida.

[4] Daniels, H. E. (1954). Saddlepoint approximations in statistics. *Ann. Math. Statist.*, **25**, 631-650.

[5] Daniels, H. E. (1983). Saddlepoint approximations for estimating equations. *Biometrika*, **70**, 89-96.

[6] Easton, G. and Ronchetti, E. (1986). General saddlepoint approximations. *J. Amer. Statist. Assoc.*, **81**, 420-430.

[7] Field C.A. (1982). Small sample asymptotics expansions for multivariate M-estimates. *Ann. Statist.*, **10** 672-689.

[8] Field, C.A. and Hampel, F.R. (1982). Small sample asymptotic distributions of M estimators of location. *Biometrika* **69** 29-46.

[9] Huber, P. (1981) *Robust Statistics.* John Wiley, New York.

[10] Lahiri, S.N. (1992a). On Bootstrapping M-estimators. *Sankhya, Ser. A* **54**, 157-170.

[11] Lahiri, S.N. (1992b). Bootstrapping M-estimators of a multiple linear regression parameter. *Ann. Statist.* **20**, 1549-1570.

[12] Qumsiyeh, M.H. (1990). Edgeworth expansions in regression models. *J. Multiv. Anal.* **35** 86-101.

[13] Reid, N. (1988). Saddlepoint methods and statistical inference. *Statistical Sciences* **3**, 213-238.

Rank CUSUM For Testing Changes In Mean

Tze-San Lee
Western Illinois University

Abstract. The cumulative sum (CUSUM) technique is widely used in industrial quality control to detect a small change in the mean. A shortcoming of the CUSUM technique is that it is not robust, namely, very sensitive to a few wild observations. A nonparametric version of the CUSUM method based on the rank statistics and its standardization is proposed which is more robust than the original CUSUM. Two examples are used to illustrate the proposed test.

1. Introduction

Many production processes subject to external stimulants may result in a change such that the mean of the process deviates from the specified target value. It is important that such a deviation could be detected as early as possible. The CUSUM technique proposed by Page [2-4] was more powerful for detecting a small change in the mean level of a continuous pocess than the conventional Shewhart's control chart. Therefore, it is widely used in the area of quality control. See, for example, Bissell [1] for a review. However, a shortcoming of the CUSUM method is that it is not robust, namely, very sentive to a few wild observations.

To overcome this shortcoming, a nonparametric version of the CUSUM method and its standardization based on the rank statistics is proposed. Although other nonparametric tests were considered before (McGilchrist-Woodyer [2], Pettitt [6], Wolfe-Schechtman [8]), the one proposed here has advantages that it is easier to implement computationally and can be visualized graphically for the slope change between sucessessive points in the sequential plot of the rank cusum as characterized in the original CUSUM method.

2. Rank CUSUM Test

Let $\{X_i\}$, $i=1,\ldots, n$ with n being given, be a sequence of independent, continuous random variables such that X_j, $j=1,\ldots, k$, has a probability distribution $F(x)$, and X_j, ,$j=k+1,\ldots, n$, has a probability distribution $F(x-\delta)$, where both k and δ are unknown with $2\le k\le n -1$ and $-\infty<\delta<\infty$. The integer k is called the change-point and δ the magnitude of change. We consider the problem of testing the null hypothesis of no change, $H_0 : \delta=0$, against the alternative of change, $H_1 : \delta>0$ (or $\delta<0$, or $\delta\neq 0$) .

Let R_i be the rank of X_i in the ordered sequence of $X_{(1)}<X_{(2)}<\ldots<X_{(n)}$. For testing $H_0 : \delta=0$ against $H_1 : \delta>0$ (or $\delta<0$) , the rank version of the CUSUM and its standardization are defined, respectively, by

$$U_1^- = -\min_{2\le q\le n-1}\{U_{1,q}\} \qquad (1)$$

$$(\text{or } U_2^- = -\min_{2\le q\le n-1}\{U_{2,q}\} \qquad (2))$$

where $U_{1,q}$ and $U_{2,q}$ are given by

$$U_{1,q} = \Sigma_{i=1,\,q}(R_i - (n+1)/2) \qquad (3)$$

and

$$U_{2,q} = U_{1,q}/(q(n-q)(n+1)/12)^{1/2} \qquad (4)$$

Rreject H_0 for large values of $U_1^- = -\min_{2\le q\le n-1}\{U_{1,q}\}$ or $U_2^- = -\min_{2\le q\le n-1}\{U_{2,q}\}$ (or reject H_0 for large values of $U_1^+ = \max_{2\le q\le n-1}\{U_{1,q}\}$ or $U_2^+ = \max_{2\le q\le n-1}\{U_{2,q}\}$).

For two-sided alternative $H_1 : \delta\neq 0$, reject H_0 for large values of $U_1^* = \max_{2\le q\le n-1}\{|U_{1,q}|\} = \max\{U_1^-, U_1^+\}$ or $U_2^* = \max_{2\le q\le n-1}\{|U_{2,q}|\} = \max\{U_2^-, U_2^+\}$. Also, an estimate of the unknown change-point is given by κ which satisfies the following

$$|U_{2,\kappa}| = \max_{2\le q\le n-1}\{|U_{2,q}|\} \qquad (5)$$

Upon a closer examination, both $U_{1,q}$ and $U_{2,q}$ are noticed to be of type of the Wilcoxon test. Due to the inherent nature of the Mann-Whitney-Wilcoxon statistics, U_1^* and U_2^*can be shown to be equivalent to the statistics K_T of Pettitt [6] and V of Schechtman [7]. Although they are equivalent, the statistics U_1^* and U_2^* are more convenient to compute than K_T and V because it only requires ranking n observations. In contrast, both K_T and V based upon the Mann-Whitney counting form require the computation of q(n-q) differences which can become unmanageable even for moderate values of q and n - q.

Also, note that the correct starting value of the index q should be from 2 rather than from 1 as used in both Pettitt [6] and Schechtman [7]. The justification is that neither k=n nor k=1,due to symmetry, can be regarded as a change-point. Another reason is that at least two points are needed to estimate the slope in the rank cusum plot as given in Section 3.

Small sample null distributions of U_2^* can be obtained by evaluating the testing statistics U_2^* for all possible arrangements of the appropriate ranks for a sample of n observations. A C-program has been designed for the personal computer to generate the exact null distribution for any sample size. However, due to the limitation of computing speed and the storage constraint of the memory space of the personal computer, only the critical values of the null distribution of sample size n = 6, 7, 8, 9, 10, and 11 are given in Table 1.

Table 1 Critical values & exact significance levels of U_2^*

	Nominal					
n	α=.10		α=.05		α=.01	
6	1.96	.10	_		_	
7	2.12	.10	_		_	
8	2.24	.08	2.31	.03	_	
9	2.20	.08	2.45	.03	_	
10	2.17	.10	2.40	.03	2.61	.008
11	2.25	.09	2.46	.03	2.74	.008

3. Application

In practical applications, the interest is often aimed at the estimation of the unknown change-point k if such a change has occurred. Just like the original CUSUM, the emphasis is on plotting the rank cusum $U_{1,q}$ or $U_{2,q}$ against q and the change-point can be visualized vividly at the point where the slope between successive points has changed dramatically. Two examples are given to illustrate the use of the rank cusum test. A C-program written to implement the rank cusum plot is available upon request from the author.

Example 1. The data given in the second row of Table 2 is taken from Pettitt [6] which are some industrial data representing the percentage of a particular material in 27 batches taken from a given source.

To demonstrate the lack of robustness of the CUSUM method, the cusum calculated from the formula $S_q = \Sigma_{i=1,q}(X_i - \bar{A})$, where $\bar{A}$ is the sample mean of all 27 observations, is given the sixth row of Table 2. The third, fourth and fifth rows of Table 2 represent the values of R_q, $U_{1,q}$ and $U_{2,q}$, respectively. As can be seen from Fig. 1(a), the "spike" value of S_q occurs at q = 7, which reflects the undue effect of the wild observation of X_8 = 17.7, and certainly is not a satisfactory estimate of the change-point. An experienced analyst, the cusum plot of S_q is highly varied before q = 16, it is much less varied after q = 16; hence the mean has probably changed at q = 16. Both of U_1^* and U_2^* give the estimate of the change-point κ = 16 since the slope has changed dramatically there (Fig.1(b)-(c)).

Table 2 The value of X_q, R_q, $U_{1,q}$, $U_{2,q}$, and S_q.

q	1	2	3	4	5
X_q	7.1	8.1	8.2	11.1	6.6
R_q	8	12	14.5	25	5
$U_{1,q}$	-6	-8	-7.5	3.5	-5.5
$U_{2,q}$	-0.07	-0.74	-0.58	0.24	-0.34
S_q	-1.33	-1.66	-1.89	0.78	-1.05
q	6	7	8	9	10
X_q	4.9	4.0	17.7	6.5	4.6
R_q	3	1	27	4	2
$U_{1,q}$	-16.5	-29.5	-16.5	-26.5	-38.5
$U_{2,q}$	-0.96	-1.63	-0.88	-1.36	-1.93
S_q	-5.58	-9.01	0.26	-1.67	-5.7
q	11	12	13	14	15
X_q	8.8	11.6	6.8	7.5	6.9
R_q	17	26	6	9.5	7
$U_{1,q}$	-35.5	-23.5	-31.5	-36	-43
$U_{2,q}$	-1.75	-1.15	-1.53	-1.75	-2.1
S_q	-5.33	-2.16	-3.79	-4.7	-6.24
q	16	17	18	19	20
X_q	8.1	9.3	7.5	10	8.7
R_q	12	21	9.5	24	16
$U_{1,q}$	-45	-38	-42.5	-32.5	-30.5
$U_{2,q}$	-2.22	-1.91	-2.19	-1.73	-1.69
S_q	-6.57	-5.7	-5.63	-4.06	-3.97
q	21	22	23	24	25
X_q	9.1	8.9	9.1	9.6	8.1
R_q	19.5	18	19.5	22	12
$U_{1,q}$	-25	-21	-15.5	-7.5	-9.5
$U_{2,q}$	-1.46	-1.31	-1.06	-0.58	-0.88
S_q	-3.12	-2.65	-1.98	-0.81	-1.14
q	26	27			
X_q	9.8	8.2			
R_q	23	14.5			
$U_{1,q}$	-0.5	0			
$U_{2,q}$	-0.06	undefined			
S_q	0.23	0			

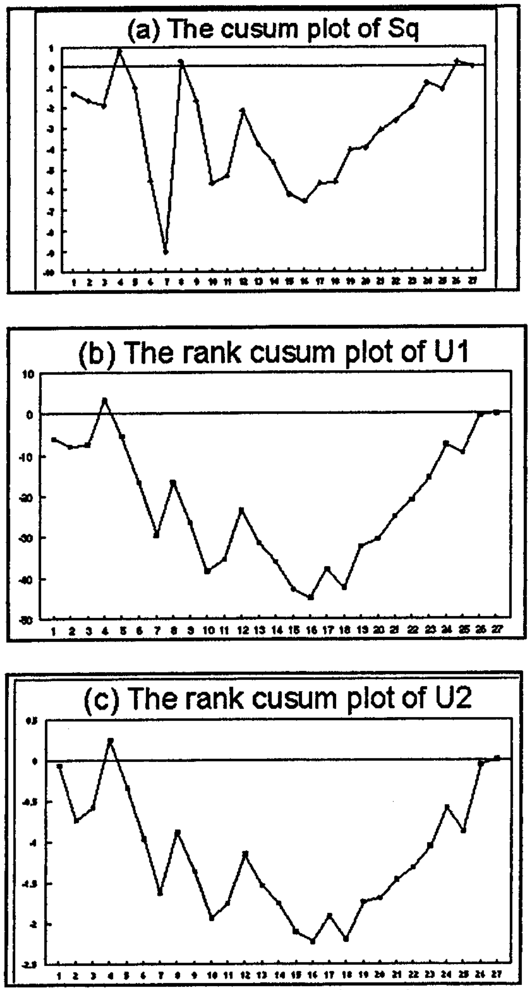
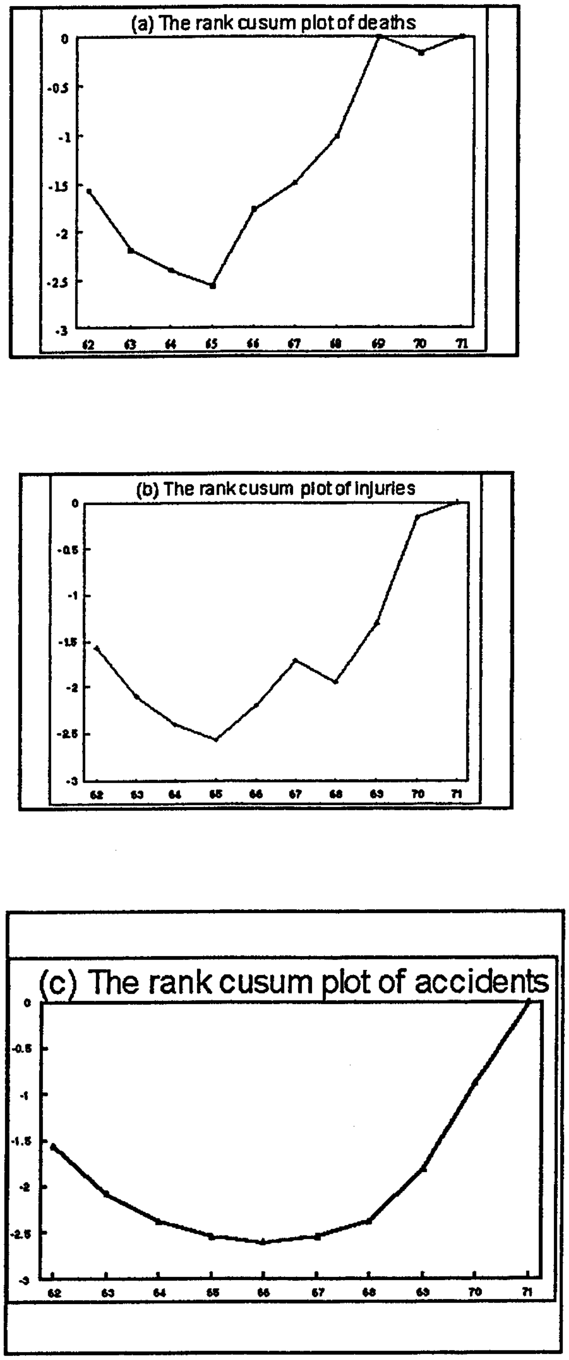


Fig. 1 The plot of cusum and its rank counterparts



Example 2. The data used here is taken from Sen-Srivastava [8], namely, the Illinois traffic data. After applying the rank cusum test (Fig. 2), the estimated change-points of deaths and injuries are the same, i.e., $\kappa=1965$ with $U_1^-=2.56$, while $\kappa=1966$ ($U_1^-=2.61$) and $\kappa=1967$ ($U_1^+=2.45$) are the change-point estimates for the data of accident and death rate. Clearly, the mean level of the deaths, injuries and accidents was increased. Only the mean level of the death rates was decreased. Also, note that all changes are significant at the level of 0.05 (Table 1 with n=10).

4. Concluding Remarks

In this paper we have presented two nonparametric tests which are the rank version of the traditional CUSUM technique. Through an example the rank cusum test is demonstrated to be more robust than the traditional cusum

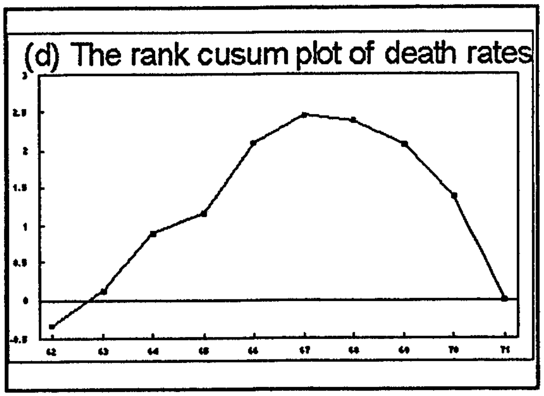


Fig. 2 The rank cusum plot of the Illinois traffic data

method. In practice, the standardized rank cusum U_2^* is recommended over the ordinary rank cusum U_1^* because finite sample null distribution of U_2^* is already constructed, but not of U_1^*. In addition, the proposed test appears not limited to the change-point problem of having at most one change. If it is visualized to have more than one change-point from the rank cusum plot, all we have to do is to split the data set into two subsets using the first change-point estimate as a dividing point and then apply the rank cusum test to each of the two subsets.

Evidently, more works are still needed to be done. For example, what is the sampling distribution of the change-point estimate κ? Without it, the confidence intervals and bounds for the change-point k can not be calculated. In practical applications, most data collected in a time order tends to be correlated. Then, the question arises: how robust is the rank cusum test when applied to the time series data?

Acknowledgements

The computation of Table 1 assisted by Mr. Lei Zhang, a graduate assistant in the Department of Mathematics of Western Illinois University, and the plot of Fig. 1 by Mr. Pete LaPorta of Armstrong Laboratory are gratefully acknowledged.

References

1. Bissell, A.F. (1969). Cusum techniques for quality control. Appl. Statist., 18, 1-30
2. McGilchrist, C.A. and Woodyer, K.D. (1975). Note on a distribution-free cusum technique. Technometrics, 17, 321-325.
3. Page, E.S. (1954). Continuous inspection schemes. Biometrika, 41, 100-114.
4. ____ (1955). A test for a change in a parameter occurring at an unknown point. Biometrika, 42, 523-527.
5. ____ (1957). On problems in which a change in parameter occurs at an unknown point. Biometrika, 44, 248-252.
6. Pettitt, A.N. (1979). A non-parametric approach to the change-point problem. Appl. Statist., 28, 126-135.
7. Schechtman, E. (1982). A non-parametric test for detecting changes in location. Comm. Statist., A11, 1475-1482.
8. Sen, A. and Srivastava, M.S. (1975). Some one-sided tests for changes in level. Technometrics, 17, 61-64.
9. Wolfe, D.A. and Schechtman, E (1984). Nonparametric statistical procedures for the changepoint problem. J. of statist. Plan. and Infer., 9, 389-396.

Perturbation Bounds for Linear Regression Problems

Bert W. Rust
Computing and Applied Mathematics Division
Building 101, Room A-238
National Institute of Standards and Technology
Gaithersburg, MD 20899
bwr@cam.nist.gov

Abstract

This paper examines errors in the estimated solution vector $\hat{\mathbf{x}}$ to the linear regression problem

$$\hat{\mathbf{y}} = \mathbf{K}\mathbf{x}^* + \hat{\epsilon} \ , \quad \mathcal{E}(\hat{\epsilon}) = \mathbf{o} \ , \quad \mathcal{E}\left(\hat{\epsilon}\hat{\epsilon}^T\right) = \mathbf{S}^2 \ ,$$

when the dominant uncertainties are the measuring errors $\hat{\epsilon}$. Backward error analysis gives the hopelessly pessimistic bound

$$\frac{\| \hat{\mathbf{x}} - \mathbf{x}^* \|_2}{\| \mathbf{x}^* \|_2} \leq \mathrm{cond}(\mathbf{S}^{-1}\mathbf{K}) \frac{\| \mathbf{S}^{-1}\hat{\epsilon} \|_2}{\| \mathbf{S}^{-1}\mathbf{K}\mathbf{x}^* \|_2} \ ,$$

by assuming the worst possible combination of random errors, an extremely unlikely occurence for nontrivial problems. A statistical treatment yields a more realistic bound on the expected uncertainty in a single element $\hat{x}_i$ which does not depend on $\mathrm{cond}(\mathbf{S}^{-1}\mathbf{K})$. Classical regression theory provides easily computable confidence intervals for the individual $\hat{x}_i$ separately.

Notation and Test Problem

Statisticians write the $m \times n$ linear regression model as

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \ , \ \ \mathcal{E}(\epsilon) = \mathbf{o} \ , \ \ \mathcal{E}\left(\epsilon\, \epsilon^T\right) = \Sigma^2 \ , \qquad (1)$$

where $\mathbf{Y}$ is a measured m-vector containing measuring errors ϵ, $\mathbf{X}$ is a known $m \times n$ matrix with $m \geq n = \mathrm{rank}(\mathbf{X})$, and β is the vector to be estimated. Numerical analysts write the linear least squares problem as

$$\rho_{LS}^2 = \min_{x \in R^n} \|b - Ax\|_2^2 \ , \qquad (2)$$

where b is the measured m-vector, A is the $m \times n$ matrix, x is the vector to be estimated, $\|b - Ax\|_2^2$ is the squared two-norm of the residual vector, and ρ_{LS}^2 is the minimum sum of squared residuals. They usually assume (but seldom state) the linear regression model

$$b = Ax^* + \delta b \ , \ \ \mathcal{E}(\delta b) = 0 \ , \ \ \mathcal{E}\left(\delta b\, \delta b^T\right) = \sigma^2 I_m \ , \qquad (3)$$

where I_m is the mth order identity matrix, and the scalar σ is unknown.

Since choosing either of the above notations would deeply offend one of the two schools, consider

$$\hat{\mathbf{y}} = \mathbf{K}\mathbf{x}^* + \hat{\epsilon} \ , \ \ \mathcal{E}(\hat{\epsilon}) = \mathbf{o} \ , \ \ \mathcal{E}\left(\hat{\epsilon}\, \hat{\epsilon}^T\right) = \mathbf{S}^2 \ , \qquad (4)$$

where $\hat{\mathbf{y}}$ is the measured m-vector, and $\mathbf{K}$ is the known $m \times n$ matrix with $\mathrm{rank}(\mathbf{K}) = n$. This notation is appropriate when linear regression is applied to systems of integral equations of the form

$$\hat{y}_i = \int_a^b K_i(\xi)x(\xi)\, d\xi + \hat{\epsilon}_i \ , \ \ i = 1, 2, \ldots, m \ , \qquad (5)$$

where the $\hat{y}_i$ are measured values, the $K_i(\xi)$ are known functions, and $x(\xi)$ is the function to be estimated. Such equations are widely used to model the effects of a measuring instrument on the thing being measured. One way to approximate $x(\xi)$ is to replace the integrals with quadrature sums, i.e.,

$$\int_a^b K_i(\xi)x(\xi)d\xi \approx \sum_{j=1}^{n} \omega_j K_i(\xi_j)x(\xi_j) \ , \qquad (6)$$

where the ω_j are prescribed quadrature coefficients and the $x(\xi_j)$ form a discrete approximation to $x(\xi)$. It is important to choose n large enough so that the quadrature errors are small relative to the $\hat{\epsilon}_i$. If the sums are substituted for the integrals in (5) and the products $\omega_j K_i(\xi_j)$ collected into a matrix $\mathbf{K}$, the result is the model (4).

A test problem capturing many of the salient features of real instrument correction problems is obtained by discretizing the Phillips [5] equation

$$y(t) = \int_{-3}^{3} K(t,\xi)x(\xi)\, d\xi \ , \ \ -6 \leq t \leq 6 \ , \qquad (7)$$

with

$$K(t,\xi) = \begin{cases} 1 + \cos\left[\frac{\pi(\xi - t)}{3}\right] \ , & | \xi - t | \leq 3 \\ & | t | \leq 6 \\ 0 \ , \ \text{otherwise} \ , & \end{cases} \qquad (8)$$

and

$$y(t) = \begin{cases} (6-\mid t \mid)\left[1+\frac{1}{2}\cos\left(\frac{\pi t}{3}\right)\right] & \\ \quad + \frac{9}{2\pi}\sin\left(\frac{\pi|t|}{3}\right), & \mid t \mid \leq 6 \\ 0\,, \ \text{otherwise}\,. & \end{cases} \qquad (9)$$

The kernel $K(t,\xi)$ is non-negative, with maximum value 2, attained on the line $t=\xi$. The solution is

$$x(\xi) = \begin{cases} 1+\cos\left(\frac{\pi\xi}{3}\right) & , \ \mid \xi \mid \leq 3 \\ 0 & , \ \text{otherwise} \end{cases} \qquad (10)$$

The functions $y(t)$ and $x(\xi)$ are plotted in Figure 1.

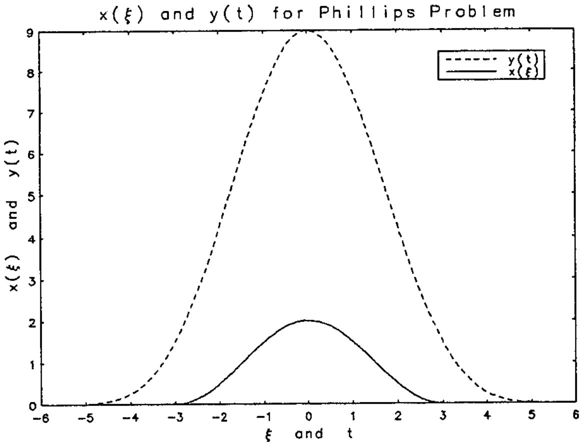


Figure 1:

Discretizing replaces continuous variables t and ξ with meshes $t_i, i=1,\ldots,m$ and $\xi_j, j=1,\ldots,n$. Choosing $m=150$ equi-spaced t_i on $-5.925 \leq t \leq 5.925$ and using an $n=121$ point trapezoidal rule on $-3.0 \leq \xi \leq 3.0$ gave

$$\mathbf{y}^* \equiv \mathbf{K}\mathbf{x}^*\,, \qquad (11)$$

where $\mathbf{x}^*$ is a 121-vector of $x(\xi_j)$ computed by (10), and $\mathbf{y}^*$ was computed by (11) rather than (9) to assure that the $\hat{\epsilon}_i$ were the only errors in the model. The $\hat{\epsilon}_i$ were obtained by random sampling from $N(\mathbf{o},\mathbf{S}^2)$ with

$$\mathbf{S} = \mathbf{diag}(s_1,\ s_2,\ \ldots,\ s_m)\,,\quad s_i = (10^{-6})y_i^*\,, \qquad (12)$$

which means that the errors in the $\hat{y}_i$ were in the 6th digit. The discretized model can thus be written

$$\mathbf{y}^* \equiv \mathbf{K}\mathbf{x}^*\,,\ \ \hat{\mathbf{y}} = \mathbf{K}\mathbf{x}^* + \hat{\boldsymbol{\epsilon}}\,,\ \ \hat{\boldsymbol{\epsilon}} \sim N(\,\mathbf{o}\,,\mathbf{S}^2\,)\,, \qquad (13)$$

and the least squares estimate

$$\hat{\mathbf{x}} = \left(\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}\right)^{-1}\mathbf{K}^T\mathbf{S}^{-2}\hat{\mathbf{y}}\,, \qquad (14)$$

computed by LINPACK subroutines DQRDC and DQRSL [2], is shown in Figure 2. The dashed curve is $x(t)$ and the jagged curve is the estimate. The large oscillations are induced by errors in the 6th digit of the $\hat{y}_i$! Such ill-conditioning is typical of regression models arising from discretized first kind integral equations.

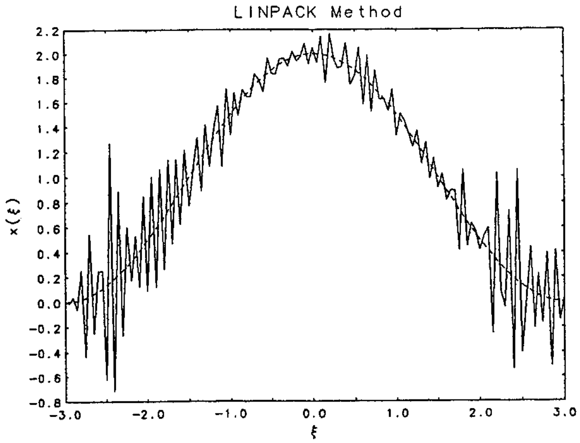


Figure 2:

Classical Perturbation Theory

To simplify the discussion in this section, let

$$\hat{\mathbf{b}} \equiv \mathbf{S}^{-1}\hat{\mathbf{y}}\,,\ \ \mathbf{A} \equiv \mathbf{S}^{-1}\mathbf{K}\,,\ \ \delta\mathbf{b} \equiv \mathbf{S}^{-1}\hat{\boldsymbol{\epsilon}}\,, \qquad (15)$$

and rewrite (13) as

$$\mathbf{b}^* = \mathbf{A}\mathbf{x}^*\,,\ \ \hat{\mathbf{b}} = \mathbf{A}\mathbf{x}^* + \delta\mathbf{b}\,,\ \ \delta\mathbf{b} \sim N(\,\mathbf{o}\,,\mathbf{I}_m\,)\,. \qquad (16)$$

The problem of interest is to find bounds for the errors in the least squares solution $\hat{\mathbf{x}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{b}}$.

The traditional approach ignores $\mathbf{x}^*$ and the statistical assumptions about $\delta\mathbf{b}$, seeking instead to bound the difference between estimates corresponding to two different $\hat{\mathbf{b}}$ vectors. One of these, $\mathbf{b}$, corresponds to the problem

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = \min = \rho_{LS}\,, \qquad (17)$$

and the other, $\mathbf{b}+\Delta\mathbf{b}$, corresponds to a perturbed problem

$$\|(\mathbf{A}+\Delta\mathbf{A})\hat{\mathbf{x}} - (\mathbf{b}+\Delta\mathbf{b})\|_2 = \min\,, \qquad (18)$$

where $\Delta\mathbf{b}$ and $\Delta\mathbf{A}$ represent the uncertainties in $\mathbf{b}$ and $\mathbf{A}$. The regression model assumes that $\mathbf{A}$ is known exactly, or at least to much higher precision than $\mathbf{b}$, but numerical analysts argue that truncation errors arising when $\mathbf{A}$ is read into a finite-accuracy computer should be taken into account. A long and intricate argument [3] leads to the following error bound:

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \leq \epsilon \left\{ \frac{2\kappa(\mathbf{A})\|\mathbf{b}\|_2 + \rho_{LS}[\kappa(\mathbf{A})]^2}{\sqrt{\|\mathbf{b}\|_2^2 - \rho_{LS}^2}} \right\} + \mathcal{O}(\epsilon^2)\,, \qquad (19)$$

where

$$\varepsilon = \max\left\{ \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2} , \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} \right\} , \qquad (20)$$

and

$$\kappa(\mathbf{A}) = \text{cond}(\mathbf{A}) = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})} = \frac{\sigma_1}{\sigma_n} \qquad (21)$$

is the *condition number* which is just the ratio of the largest to the smallest singular value of **A**.

While numerical analysts are fascinated by the truncation $\Delta\mathbf{A}$, people who actually make measurements usually insist on a computer arithmetic with enough precision to render such perturbations negligible in comparision to the measurement errors. When the Computer Acquisition Committee at the National Bureau of Standards was writing specifications for a new computer in 1984, some members insisted on a machine with 64-bit single precision because 32-bit machines give only 6 to 7 digits of precision, and they routinely measured things better than that. Accordingly, let $\Delta\mathbf{A} = \mathbf{0}$. This leads to the more easily obtained [6] bound

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} \le \text{cond}(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|_2}{\|\mathbf{b}\|_2} , \qquad (22)$$

which also depends strongly on cond(**A**).

Assessing the Classical Bound

The bound (22) is computable, but it does not relate a computed estimate to $\mathbf{x}^*$. To obtain such a result, let

$$\mathbf{b} = \mathbf{b}^* = \mathbf{A}\mathbf{x}^* , \;\; \Delta\mathbf{b} = \delta\mathbf{b} \sim \mathcal{N}(\mathbf{o}, \mathbf{I}_m) , \qquad (23)$$

and replace problems (17) and (18) with

$$\|\mathbf{A}\mathbf{x}^* - \mathbf{b}^*\|_2 = \min = 0 , \;\; \|\mathbf{A}\hat{\mathbf{x}} - (\mathbf{b}^* + \delta\mathbf{b})\|_2 = \min . \qquad (24)$$

The bound (22) then becomes

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} \le \text{cond}(\mathbf{A}) \frac{\|\delta\mathbf{b}\|_2}{\|\mathbf{A}\mathbf{x}^*\|_2} , \qquad (25)$$

which is not practicable because it depends on $\mathbf{x}^*$. But $\mathbf{x}^*$ is known for the test problem, and this provides a means for evaluating the perturbation bound. To restore the original notation, substitute (15) into (25) to obtain

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2}{\|\mathbf{x}^*\|_2} \le \text{cond}(\mathbf{S}^{-1}\mathbf{K}) \frac{\|\mathbf{S}^{-1}\hat{\epsilon}\|_2}{\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2} , \qquad (26)$$

where

$$\text{cond}(\mathbf{S}^{-1}\mathbf{K}) = \frac{\sigma_{\max}(\mathbf{S}^{-1}\mathbf{K})}{\sigma_{\min}(\mathbf{S}^{-1}\mathbf{K})} = \frac{\sigma_1}{\sigma_n} . \qquad (27)$$

Multiplying (26) by $\|\mathbf{x}^*\|_2$ and squaring both sides gives

$$\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2 \le \frac{\left[\text{cond}(\mathbf{S}^{-1}\mathbf{K})\right]^2 \|\mathbf{x}^*\|_2^2}{\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2^2} \|\mathbf{S}^{-1}\hat{\epsilon}\|_2^2 . \qquad (28)$$

Since both sides are non-negative functions of the random vector $\hat{\epsilon}$, it follows that

$$\mathcal{E}\left(\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2\right) \le \frac{\left[\text{cond}(\mathbf{S}^{-1}\mathbf{K})\right]^2 \|\mathbf{x}^*\|_2^2}{\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2^2} \mathcal{E}\left(\|\mathbf{S}^{-1}\hat{\epsilon}\|_2^2\right) . \qquad (29)$$

It follows from (13) that $\mathbf{S}^{-1}\hat{\epsilon} \sim \mathcal{N}(\mathbf{o}, \mathbf{I}_m)$ which implies $\|\mathbf{S}^{-1}\hat{\epsilon}\|_2^2 \sim \chi^2(m)$, so $\mathcal{E}\left(\|\mathbf{S}^{-1}\hat{\epsilon}\|_2^2\right) = m$. Therefore

$$\mathcal{E}\left(\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2\right) \le \frac{m\left[\text{cond}(\mathbf{S}^{-1}\mathbf{K})\right]^2 \|\mathbf{x}^*\|_2^2}{\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2^2} , \qquad (30)$$

which relates $\hat{\mathbf{x}}$ to $\mathbf{x}^*$, but with the elements of $|\hat{\mathbf{x}} - \mathbf{x}^*|$ muddled together. To clarify, define $|\Delta x|_{rms}$ by

$$|\Delta x|_{rms}^2 \equiv \mathcal{E}\left(\frac{1}{n}\sum_{j=1}^{n} |\hat{x}_j - x_j^*|^2\right) = \frac{1}{n}\mathcal{E}\left(\|\hat{\mathbf{x}} - \mathbf{x}^*\|_2^2\right) , \qquad (31)$$

so by (30),

$$|\Delta x|_{rms} \le \left(\sqrt{\frac{m}{n}}\right) \text{cond}(\mathbf{S}^{-1}\mathbf{K}) \frac{\|\mathbf{x}^*\|_2}{\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2} . \qquad (32)$$

The quantity $|\Delta x|_{rms}$ is the expected root mean squared absolute error for the components of $\hat{\mathbf{x}}$. The test problem has $\|\mathbf{x}^*\|_2 = 13.82$, $\sigma_1(\mathbf{S}^{-1}\mathbf{K}) = 3.3950 \times 10^9$, and $\sigma_{121}(\mathbf{S}^{-1}\mathbf{K}) = 1.1610$. Thus $\text{cond}(\mathbf{S}^{-1}\mathbf{K}) = 2.924 \times 10^9$, and by (12),

$$\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^* = \mathbf{S}^{-1}\mathbf{y}^* = \left(10^6, 10^6, \ldots, 10^6\right)^T , \qquad (33)$$

so $\|\mathbf{S}^{-1}\mathbf{K}\mathbf{x}^*\|_2 = 1.225 \times 10^7$. Substituting these values into (32) gives $|\Delta x|_{rms} \le 3.67 \times 10^3$, a wildly pessimistic bound. Figure 3 gives a componentwise plot of the actual errors $\hat{\mathbf{x}} - \mathbf{x}^*$ with the true values of $\pm|\Delta x|_{rms} = \pm 0.302$ plotted as dashed lines.

The classical bound is hopelessly pessimistic because it does not take the random nature of the errors into account. Starting with a measured **b** and corresponding solution **x**, it considers all measured vectors $\mathbf{b} + \delta\mathbf{b}$ with $\|\delta\mathbf{b}\|_2 \le \|\Delta\mathbf{b}\|_2$. These vectors define corresponding solutions $\hat{\mathbf{x}} = \mathbf{x} + \delta\mathbf{x}$, and to make the bound hold with certainty for all $\mathbf{b} + \delta\mathbf{b}$, it assumes the worst possible combination of the 121 perturbations $\delta\mathbf{b}$. When the errors are drawn randomly, the probability of such a combination is negligibly small.

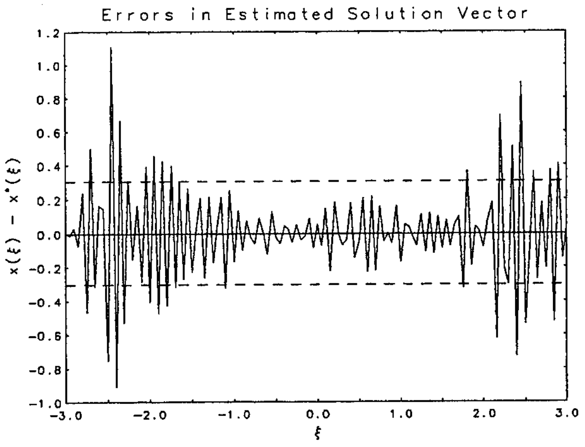


Figure 3:

Statistical Perturbation Bounds

A more reasonable bound can be obtained by considering the statistical properties of the errors. By (13),

$$(\hat{\mathbf{x}}-\mathbf{x}^*) \sim N\left[\mathbf{o}, \left(\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}\right)^{-1}\right] , \qquad (34)$$

so

$$(\hat{\mathbf{x}}-\mathbf{x}^*)^T\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}(\hat{\mathbf{x}}-\mathbf{x}^*) \sim \chi^2(n) , \qquad (35)$$

whence

$$\mathcal{E}\left\{(\hat{\mathbf{x}}-\mathbf{x}^*)^T\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}(\hat{\mathbf{x}}-\mathbf{x}^*)\right\} = n . \qquad (36)$$

Now consider the singular value decomposition

$$\mathbf{S}^{-1}\mathbf{K} = \mathbf{U}\begin{pmatrix}\boldsymbol{\Sigma}\\ \mathbf{0}\end{pmatrix}\mathbf{V}^T , \quad \boldsymbol{\Sigma} = \mathbf{diag}(\sigma_1,\sigma_2,\ldots,\sigma_n) ,$$
$$\mathbf{U}^T\mathbf{U} = \mathbf{I}_m , \quad \mathbf{V}^T\mathbf{V} = \mathbf{I}_n , \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n . \qquad (37)$$

Substituting into (36) and simplifying gives

$$\mathcal{E}\left\{\sum_{j=1}^{n}\sigma_j^2\left[\mathbf{V}^T(\hat{\mathbf{x}}-\mathbf{x}^*)\right]_j^2\right\} = n , \qquad (38)$$

and, since σ_n is the minimum singular value,

$$\sigma_n^2\mathcal{E}\left\{\sum_{j=1}^{n}\left[\mathbf{V}^T(\hat{\mathbf{x}}-\mathbf{x}^*)\right]_j^2\right\} \leq n . \qquad (39)$$

Dividing through by σ_n^2 gives

$$\mathcal{E}\left\{\sum_{j=1}^{n}\left[\mathbf{V}^T(\hat{\mathbf{x}}-\mathbf{x}^*)\right]_j^2\right\} = \mathcal{E}\left\{\|\mathbf{V}^T(\hat{\mathbf{x}}-\mathbf{x}^*)\|_2^2\right\} \leq \frac{n}{\sigma_n^2} . \qquad (40)$$

The two-norm is invariant with orthogonal rotations, so

$$\mathcal{E}\left\{\|\hat{\mathbf{x}}-\mathbf{x}^*\|_2^2\right\} \leq \frac{n}{\sigma_n^2} , \qquad (41)$$

whence, by (31),

$$|\Delta x|_{rms} \leq \frac{1}{\sigma_n} . \qquad (42)$$

This bound is computable without knowing $\mathbf{x}^*$, and *it does not depend on* $\mathrm{cond}(\mathbf{S}^{-1}\mathbf{K})$. For the test problem, $|\Delta x|_{rms} \leq 0.861$, which exceeds the true value by a factor of only 2.85.

Confidence Intervals

Both the classical and statistical perturbation analyses are rendered moot by confidence interval calculations. If $\hat{\mathbf{x}}$ is the least squares solution for the model (13), then

$$\hat{\mathbf{x}} \sim N\left[\mathbf{x}^*, \left(\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}\right)^{-1}\right] , \qquad (43)$$

so the variances of the invidual $\hat{x}_j$ are given by

$$\mathrm{V}(\hat{x}_j) = \mathbf{e}_j^T\left(\mathbf{K}^T\mathbf{S}^{-2}\mathbf{K}\right)^{-1}\mathbf{e}_j^T , \quad j = 1,2,\ldots,n , \qquad (44)$$

where $\mathbf{e}_j$ is the unit vector with 1 as the jth element. For any probability α ($0 < \alpha < 1$), if κ is chosen to satisfy

$$\frac{1}{\sqrt{2\pi}}\int_{-\kappa}^{+\kappa}\exp\left(-\frac{\eta^2}{2}\right)d\eta = \alpha , \qquad (45)$$

then

$$\Pr\left\{\left[\hat{x}_j - \kappa\sqrt{\mathrm{V}(\hat{x}_j)}\right] \leq x_j^* \leq \left[\hat{x}_j + \kappa\sqrt{\mathrm{V}(\hat{x}_j)}\right]\right\} = \alpha . \qquad (46)$$

The κ-value for $\alpha = .95$ is $\kappa = 1.96$. Figure 4 shows the 95% confidence bounds for the test problem. The dashed line is the true solution and the jagged lines connect the upper and lower bounds for the individual $\hat{x}_i$.

If $\mathbf{S}^2 = s^2 I_m$, with s unknown, then the estimate $\hat{s}^2 = (m-n)^{-1}\rho_{LS}^2$ can be used to construct confidence intervals, though the relation between κ and α will be different from (45). If the $\hat{\epsilon}$-distribution is unknown, confidence intervals can be constructed from the Chebeyshev inequality. Though wider than those for normally distributed errors, these intervals are often orders of magnitude smaller than the $\pm|\Delta x|_{rms}$ bounds from classical perturbation theory.

The keynote speaker [7] pointed out that the variance matrix for $\hat{x}_j$ was known to Gauss, and that modern least squares algorithms could easily compute it by inverting an upper triangular matrix formed in solving

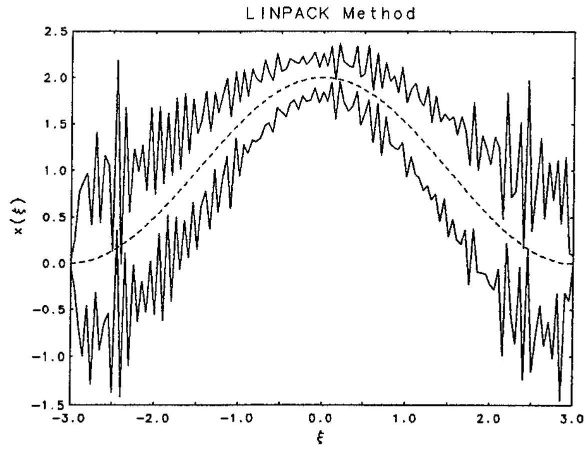


Figure 4:

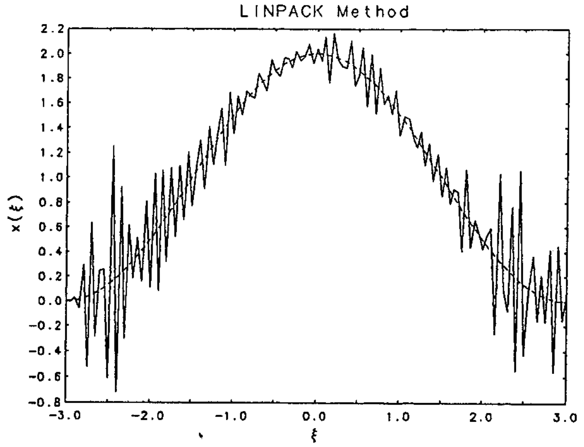


Figure 5:

for $\hat{\mathbf{x}}$. Unfortunately, the least squares subroutines in the widely used LINPACK [2] and LAPACK [1] collections do not return confidence intervals, or even the variance matrix. The LINPACK manual describes how to compute variances from a reduced matrix returned by subroutine SQRDC, but the LAPACK manual is silent on the subject, and neither mentions confidence intervals, concentrating instead on the classical perturbation bounds. Secondary sources, which use these collections, have continued this preoccupation with what are essentially useless bounds. They also continue to propagate misinformation about the condition number. For example, the textbook of Kahaner, et. al [4] states that:

> One useful interpretation of the condition number is that its logarithm approximates the number of digits which will be lost while solving $Ax = b$. Thus if $\mathrm{cond}(A) = 10^5$ and if machine epsilon is 10^{-8}, then the best we can expect is that the solution will be accurate to about three digits.

The estimate in Figure 2 was calculated in double precision with $\epsilon_{\text{mach}} = 2.22 \times 10^{-16}$, and since $\mathrm{cond}(\mathrm{S}^{-1}\mathrm{K}) = 2.92 \times 10^9$, the above reasoning would indicate that the computed $\hat{\mathbf{x}}$ is accurate to 6 digits. But consider the same calculation in single precision with $\epsilon_{\text{mach}} = 1.19 \times 10^{-7}$ and $\mathrm{cond}(\mathrm{S}^{-1}\mathrm{K}) = 2.93 \times 10^9$. According to the conventional wisdom, a computed estimate should not contain any digits of accuracy. The actual single precision estimate is shown in Figure 5. The slight differences from the double precision estimate are difficult to see by comparing the two plots. The rms average difference between the two estimates is 0.0033 which is almost 100 time smaller than the $|\Delta x|_{rms}$ for either estimate, so in practice, either estimate would serve equally well. Clearly the condition number is not always a good indicator of the accuracy of the estimate.

Acknowledgements

I would like to thank Dr. W.R. Burrus for pointing out that the classical perturbation bounds are usually too pessimistic to be useful and Dr. R.F. Boisvert for many useful suggestions for improving the manuscript.

References

[1] Anderson, E., Bai, Z., Bischof, C., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Ostrouchov, S., and Sorensen, D. (1992) *LAPACK User's Guide*, SIAM, Philadelphia.

[2] Dongarra, J.J., Moler, C.B., Bunch, J.R., and Stewart, G.W. (1979) *LINPACK Users' Guide*, SIAM, Philadelphia, Chapt. 9.

[3] Golub, G.H. and Van Loan, C.F. (1989) *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Chapt. 5.

[4] Kahaner, D., Moler, C., and Nash, S. (1989) *Numerical Methods and Software*, Prentice Hall, Englewood Cliffs, Chapt. 3.

[5] Phillips, D.L. (1962) *J. Assoc. Comput. Mach.*, **9**, pp. 84-97.

[6] Stewart, G.W. (1973) *Introduction to Matrix Computations*, Academic Press, New York, Chapt. 5.

[7] Stewart, G.W. (1994) *This Volume.*

Tailoring Nonlinear Least Squares Algorithms for the Analysis of Compartment Models

David M. Allen
Department of Statistics
University of Kentucky

Abstract

Compartment models are widely used in pharmacokinetics. Our objective is to fit compartment models to data. General numerical optimization methods frequently perform poorly for this purpose. A good book on numerical optimization, such as Dennis and Schnabel's[3] describes multiple techniques and discusses the advantages and disadvantages of each. In order to implement these methods in a software product, one must make a number of decisions. For example, should a line search method or a trust region method be used? How should variables on widely different scales be handled? In general, these are questions without clear-cut answers.

Compartment models are defined by linear differential equations. Consequently, compartment models have a particular structure. We have tailored general optimization methods to exploit this structure. Through study and experimentation we have found workable answers to the questions posed above.

1 Introduction

Compartment models are illustrated with a study of gold kinetics. I will give some background for the study, the kinetic diagram, the differential equations, and the data. The scaling problem is described, and a method for dealing with it is presented. A variety of algorithms are described our preference is stated.

2 An example model

This example from Gerber, *etal*[4] deals with gold kinetics. The effects of aurothiomalate therapy last far beyond the time where there are measurable serum levels. However, whole-body radiation counts can made over any interval of time. In this study, serum levels and whole-body counts are simultaneously fit to a two compartment model. We assume the blood serum is a compartment, and the remainder of the body is a compartment. The compartments and flows are depicted in Figure 1. The parameters k21, k12, and k01 are called rate constants.

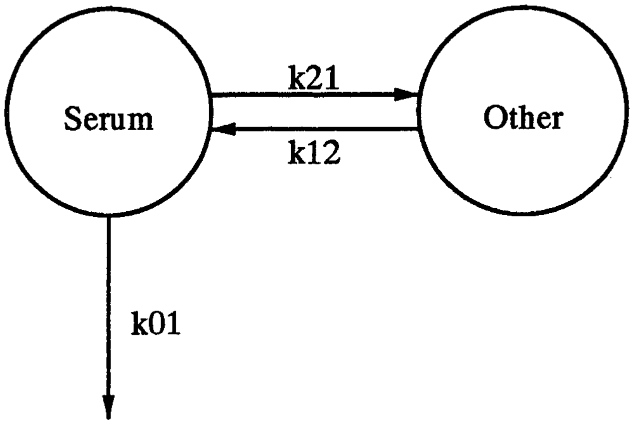


Figure 1: Model for gold kinetics study

Aurothiomalate is injected into the blood serum. At several values of elapsed time, two types of observations are made: the concentration in the serum and a radioactive count on the whole body. Dose 1 is the initial value in the serum compartment in units of concentration. Dose 2 is the initial value of the sum of the two compartments in units of counts. For theoretical discussions, the parameters are in an indexed vector θ. We use the following correspondence: $k21 = \theta_1$, $k12 = \theta_2$, $k01 = \theta_3$, Dose $1 = \theta_4$, and Dose $2 = \theta_5$.

The differential equations associated with the kinetic diagram are

$$\begin{pmatrix} \dot{P}_{1j}(t) \\ \dot{P}_{2j}(t) \end{pmatrix} = \begin{pmatrix} -\theta_1 - \theta_3 & \theta_2 \\ \theta_1 & -\theta_2 \end{pmatrix} \begin{pmatrix} P_{1j}(t) \\ P_{2j}(t) \end{pmatrix}$$

The subscript j is used to distinguish between solutions resulting from different initial conditions. Specifically, $\begin{pmatrix} P_{1j}(t) & P_{2j}(t) \end{pmatrix}^T$ is the solution when $\begin{pmatrix} P_{1j}(0) & P_{2j}(0) \end{pmatrix}^T$ is the jth elementary vector. $P_{ij}(t)$ is the proportion of material that goes from compartment j to compartment i in the time interval $(0, t)$. For the deterministic form of the model we need only the solution for $j = 1$ because the dose is administered in the first compartment. The solution is

$$\lambda_1, \lambda_2 = \frac{-(\theta_1 + \theta_2 + \theta_3) \pm \sqrt{(\theta_1 + \theta_2 + \theta_3)^2 - 4\theta_2\theta_3}}{2}$$

$$P_{11}(t) = \frac{-\lambda_1(\theta_3 + \lambda_2)\exp(\lambda_1 t) + \lambda_2(\theta_3 + \lambda_1)\exp(\lambda_2 t)}{\theta_3(\lambda_2 - \lambda_1)}$$

$$P_{.1}(t) = \frac{(\theta_3 + \lambda_2)\exp(\lambda_1 t) - (\theta_3 + \lambda_1)\exp(\lambda_2 t)}{\lambda_2 - \lambda_1}$$

where $P_{.1}(t) = P_{11}(t) + P_{21}(t)$.

The analytical solutions given above are to make the example precise and self-contained. In practice, a computer can solve the required differential equations and also find the derivatives of the solutions with respect to the parameters.

The data were read from figures in Gerber *etal* by Uno *etal*[5], and are given in Table 1. The expected values of the observations y_i are

$$f_i(\theta) = \begin{cases} \theta_4 \times P_{11}(t_i) & \text{for } i \leq 7 \\ \theta_5 \times (P_{11}(t_i) + P_{21}(t_i)) & \text{for } i \geq 8. \end{cases}$$

The method of estimation is to find the value of the parameter vector θ so that

$$Q(\theta) = \sum_{i=1}^{n} \left(\frac{y_i^\lambda - f_i^\lambda(\theta)}{\lambda} \right)^2 \qquad (1)$$

is minimized. The parameter λ is specified by the data analyst to stabilize the variances of the y_i. For example, $\lambda = 0.5$ gives the square root transformation, and $\lambda = 0.0$ gives the logarithmic transformation. See Box and Cox[2] for theory and strategies of choosing λ.

i	Site	t_i	y_i
1	serum	1.06	354.0
2	serum	2.13	284.0
3	serum	3.19	238.0
4	serum	4.26	200.0
5	serum	5.11	175.0
6	serum	6.17	145.0
7	serum	7.23	128.0
8	body	0.00	100.0
9	body	3.11	87.23
10	body	5.19	79.79
11	body	14.53	60.64
12	body	21.79	54.26
13	body	41.51	46.81
14	body	62.26	44.68
15	body	97.55	41.49
16	body	174.34	36.17
17	body	215.85	34.04

Table 1: The data

3 Dealing with the scaling problem

The two classes of parameters, rate constants, and initial values, have vastly different scales. Ordinary nonlinear regression algorithms can be very slow to converge.

The problems can be demonstrated using a model simpler than the one presented in the preceding section

$$y_i = D\frac{k_a}{-k_a + k_e}(\exp(-k_a t_i) - \exp(-k_e t_i)) + \epsilon_i.$$

Figure 2 contains two response curves for this model. There are two rate constants, k_a and k_e, and they are the same for each curve. The initial value, D, for the lower curve is one tenth the initial value in the upper curve. Artificial data are taken from the upper curve with no error term. Parameters of the lower curve are used as starting values for a common algorithm. Convergence is very slow even though two of the three parameter estimates are correct.

We return our attention to the gold kinetics study. When $\lambda = 1.0$, θ_4 and θ_5 are conditionally linear parameters. Bates and Watts[1] discuss handling conditionally linear parameters. The method is to alternate between normal iterations and off iterations where the rate constants are fixed.

I propose doing exactly the same thing even when $\lambda \neq 1.0$. While these off iterations are not linear problems, the process allows the initial value parameters to

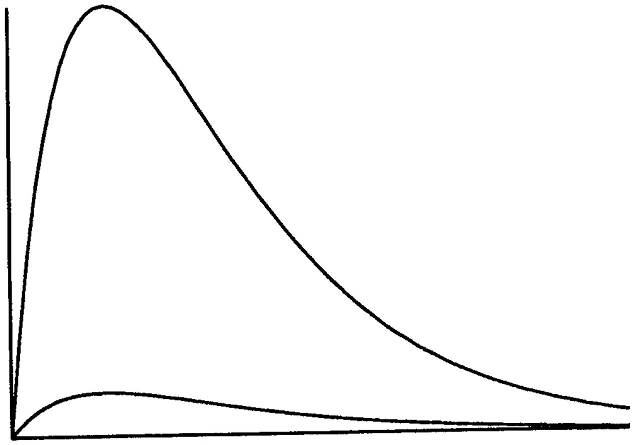


Figure 2: A pathological example

partially adjust to the current values of the rate constants. Experience with several examples has shown that speed of convergence can be dramatically improved using this technique.

4 Choice of method

Newton's method for minimizing the objective function (1) is probably best, but because it requires second derivatives, is infrequently used. A modified Gauss-Newton method is nearly always used. The question is which modification should we use?

We use the following notation to describe possible estimators:

$$\begin{aligned} J(\theta) &= \frac{\partial}{\partial\theta} Q(\theta) \\ &= \left(f_i(\theta)^{\lambda-1} \frac{\partial}{\partial\theta_j} f_i(\theta) \right) \\ r(\theta) &= \left(\frac{y_i^\lambda - f_i^\lambda(\theta)}{\lambda} \right) \end{aligned}$$

All nonlinear regression algorithms are iterative. At each iteration the current values of the estimates are updated by adding an adjustment vector. Possible formulas for the adjustment vectors are:

$$(J(\theta)^T J(\theta))^{-1} J(\theta)^T r(\theta) \tag{2}$$

$$(J(\theta)^T J(\theta) + \gamma I)^{-1} J(\theta)^T r(\theta) \tag{3}$$

$$\sigma(J(\theta)^T J(\theta))^{-1} J(\theta)^T r(\theta) \tag{4}$$

$$\sigma(J(\theta)^T J(\theta) + \gamma I)^{-1} J(\theta)^T r(\theta) \tag{5}$$

Expression (2) is the Gauss-Newton formula which will often not converge. Expression (3) is the Levenberg-Marquardt formula. This method is popular, and many different strategies for choosing γ have been proposed. A combination of expression (2) and expression (3) is called the trust region method and is detailed in Dennis and Schnabel[3]. Expression (4) is the line search formula and is used with a strategy for choosing σ.

I prefer a line search algorithm, backtracking with cubic interpolation when required. This is also detailed in Dennis and Schnabel. For the search direction, I use expression (5) with γ fixed at some small number. The search parameter σ adjusted at each iteration using cubic interpolation. Fixing γ to be positive avoids having to check if $J(\theta)$ is singular. The nature of compartment models is such that $Q(\theta)$ and its directional derivatives along the search line are easy to compute. The cubic interpolation provides an intelligent update.

References

[1] D. M. Bates and D. G. Watts. *Nonlinear Regression Analysis & Its Applications.* John Wiley & Sons, Inc., New York, New York, 1988.

[2] George E. P. Box and David R. Cox. An analysis of transformations. *Journal of the Royal Statistical Society, Series B*, 26:211–252, 1964.

[3] J. E. Dennis, Jr. and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1983.

[4] R. C. Gerber, H. E. Paulus, R. I. Jennrich, R. Bluestone, W. H. Blahd, and C. M. Pearson. Gold kinetics following aurothiomalate therapy: Use of a whole body radiation counter. *J. Lab. Clin. Med.*, 83(5):778–789, 1974.

[5] F. K. Uno, M. L. Ralston, R. J. Jennrich, and P.F. Sampson. Test problems from pharmacokinetic literature requiring fitting models defined by differential equations. BMDP Statistical Software 61, Department of Biomathematics, UCLA, 1979.

Characterizing Hierarchical Model Behavior

Beth L. Chance*and Martin T. Wells
Statistics Center, Cornell University
Ithaca, NY 14853

ABSTRACT

Extending results of Dawid (1973), O'Hagan (1979), Meeden & Isaacson (1977), and Angers & Berger (1991), we develop a general theory of model behavior for different distributional assumptions in a hierarchal model in the presence of outlying data. The score function of a density allows characterization of densities into four groups based on their tail behavior. Using convolution theory, we characterize the behavior of a location parameter estimator in a hierarchical model depending on the group membership of the densities involved. These results extend to multivariate distributions under the assumption of exchangeability. Using mixture distributions (Andrews and Mallows, 1974) we implement a Gibbs Sampler for prototypes from these groups. This theory indicates the model behavior for most commonly used distributions, including a variation of the multivariate Laplace.

1 INTRODUCTION

We are interested in the sensitivity of hierarchical models to the distributions specified for each level. If we assume a two level model,

$$\begin{aligned} Y|\theta,\sigma^2 &\sim f_{Y|\theta,\sigma^2}(y|\theta,\sigma^2) \\ \theta|\mu,\tau &\sim f_{\Theta|\mu,\tau}(\theta|\mu,\tau), \end{aligned}$$

we can estimate the unknown parameters θ using Empirical or Hierarchical Bayes methodology. We would like some method for knowing how the estimate of θ will behave for different structural assumptions on f_Y and f_Θ. It is well known that conjugate densities can lead to undesirable behavior (e.g. Lindley & Smith, 1972) as the data and the prior information become discrepant, i.e. $|Y-\mu| \to \infty$, by always compromising between the likelihood and prior. This has led to research on the behavior of the posterior mean for nonconjugate densities as $|Y-\mu| \to \infty$. Assuming θ is a location parameter, Dawid (1973) and O'Hagan (1979) derived conditions such that the posterior tends to the prior, thus rejecting the information from the likelihood. Reversing the conditions, the posterior behaves as the likelihood. Hill (1974) extended these results to the multivariate setting. Sansó and Pericchi (1992) examined behavior for a normal likelihood and Laplace prior, finding that the posterior mean tends to $y-c$ where c is some constant, and thus the prior exerts *bounded influence.* Angers and Berger (1991) examined the multivariate behavior for a Cauchy prior. Meeden and Isaacson (1977) developed similar theory for f_Y an exponential family and θ the canonical parameter which was extended by Pericchi, Sansó, and Smith (1993) to expectation parameters. Along similar lines, Lucas (1993) developed conditions for posterior normality when both densities belong to the Box-Tiao family.

We refer to these results as "what-if" asymptotics. They indicate how the model behaves as $|Y-\mu| \to \infty$, such as ignoring either the prior or likelihood, always compromising between them, or exhibiting bounded influence. Our aim is to develop a general theory that will describe the model behavior for arbitrary parametric forms of the likelihood and prior when θ is a location parameter. To accomplish this, we use a scheme for classifying densities into disjoint classes based on tail behavior, and demonstrate that the relative ordering of the tails determines posterior behavior. Thus, the problem of selecting densities for each level of a hierarchical model simplifies to determining class membership. We extend the results to the multivariate setting and additional levels in the hierarchy. While research in this area generally assumes the variance components are known, our results also indicate the behavior of the posterior distributions of σ^2 and τ for general priors on these scale

*Research supported by a National Defense Science and Engineering Graduate Fellowship

parameters. We also demonstrate a Gibbs Sampling implementation that immediately provides estimates of the desired posterior distributions for prototypes from each of these classes, and thus provides information for all densities in that class.

2 DENSITY CLASSIFICATION

To classify a density's tail behavior, we utilize the *negative log rate*, $\mathrm{NLR}_f(x) = -\frac{d}{dx}\log f(x)$. This is equal to minus the *score* function and will be applied to likelihoods and priors. Using a classification scheme adapted from Gomez-Villegas and Main (1992), we classify a density as

- Very Light if $\mathrm{NLR}_f(x) \to \infty$
- Light if $\mathrm{NLR}_f(x) \to c, 0 < c < \infty$
- Medium-Heavy if $\mathrm{NLR}_f(x) \to 0$
  - Medium if $x\mathrm{NLR}_f(x) \to \infty$
  - Heavy if $x\mathrm{NLR}_f(x) \to c, c < \infty$,

where all limits are as $x \to \infty$. This scheme agrees with our intuition by classifying a Normal density as Very Light, a Laplace density as Light, and the t as Heavy. We see that the tail of f_1 is heavier that the tail of f_2 if $\lim_{x\to\infty} \mathrm{NLR}_{f_1}(x) < \lim_{x\to\infty} \mathrm{NLR}_{f_2}(x)$. As shown below, this tail characteristic determines if our estimates will compromise or ignore the information from the densities involved.

3 CONVOLUTION THEORY

Using ideas from convolution theory, we obtain a theory that determines posterior behavior based on the relative NLR's of the likelihood and the prior. Since we are assuming θ is a location parameter, $f_{Y|\theta}(y|\theta)$ can be rewritten as $f^*_{Y-\theta}(y-\theta)$ for some pivot density f^*, and the marginal density $\int f^*_{Y-\theta}(y-\theta)\pi(\theta)d\theta$ is the convolution of f^* and π. Berman (1992) developed theory for the behavior of this convolution as $y \to \infty$. We have extended this theory so that it may be applied to our hierarchical models. Below is our main result, see Chance (1994) for details.

Theorem 1 *Suppose NLR_π is a regularly varying function, $NLR_f > 0$, $g(t)$ has finite expectation, and*

$$\limsup_{y\to\infty} NLR_\pi(y) < \liminf_{y\to\infty} NLR_f(y)$$

then

$$\int g(y-\theta)f^*_{Y-\theta}(y-\theta)\pi_\Theta(\theta)d\theta$$

$$\sim \pi_\Theta(y)\int g(\theta)f^*_{Y-\theta}(\theta)e^{\theta NLR_\pi(y)}d\theta$$

for $y \to \infty$, when $\int_{-\infty}^{\infty} e^{tNLR_\pi(y)}f^*_{Y-\theta}(t)dt < \infty$, and $\int g(t)f^*_{Y-\theta}(t)e^{\theta NLR_\pi(x)}dt < \infty$.

When $g(t) = 1$, this tells us that when we have an extreme data value, the marginal behaves as the prior, evaluated at the data point, times a correction factor. Note, assuming NLR_π regularly varying is not a very restrictive assumption since taking the logarithm suitably dampens commonly used density functions. Applying Theorem 1 with $g(y-\theta) = 1$ and $g(y-\theta) = y-\theta$, we see that the posterior expectation of $g(y-\theta)$ behaves as:

$$\begin{aligned} E_{\Theta|y}(g(y-\Theta)|y) &= \frac{\int g(y-t)f^*(y-t)\pi(t)dt}{\int f^*(y-t)\pi(t)dt} \\ &\sim \frac{\int g(t)f^*(t)e^{t\mathrm{NLR}_\pi(y)}dt}{\int f^*(t)e^{t\mathrm{NLR}_\pi(y)}dt}. \end{aligned}$$

We can further manipulate the equations to obtain an expression for the posterior density:

$$\begin{aligned} p_{\Theta|y}(\theta|y) &\sim \frac{f^*(y-\theta)e^{(y-\theta)\mathrm{NLR}_\pi(y)}}{\int_{-\infty}^{\infty} f^*(y-\theta)e^{(y-\theta)\mathrm{NLR}_\pi(y)}d\theta} \\ &= \frac{f^*(y-\theta)e^{-\theta\mathrm{NLR}_\pi(y)}}{\int_{-\infty}^{\infty} f^*(y-\theta)e^{-\theta\mathrm{NLR}_\pi(y)}d\theta}. \end{aligned}$$

Thus, Theorem 1 directly implies that when the prior has the heavier tail, the marginal density behaves as the prior density times a correction factor and the posterior as the pivot density times a correction factor as $y \to \infty$. When $\mathrm{NLR}_\pi(y) \to 0$, the marginal behaves as the prior, a result closely related to Brown's (1988) heuristic, and the posterior mean of $g(y-\theta)$ goes to $\int g(\theta)f^*(\theta)d\theta$. When m is an indicator function we see that the posterior distribution tends to the invariant distribution of the pivot $Y-\theta$, and $E(y-\theta|y) \to E_{f^*}(y-\theta) = 0$, that is $E(\theta|y) \to y$. This is the result given by Dawid (1973) and O'Hagan (1979). When the prior is a Light tailed density, we can often evaluate the correction factor exactly.

Example Let $f(y|\theta) \sim N(\theta, \sigma^2)$ and $\pi(\theta) \sim DE(0, \tau^2)$. Since $\mathrm{NLR}_{f^*}(y) = y$ and $\mathrm{NLR}_\pi(y) = \lambda = \frac{\sqrt{2}}{\tau}$, which is regularly oscillating, the conditions of the theorem are met. Applying the result,

$$\begin{aligned} p_{\Theta|y}(\theta|y) &\sim \frac{f^*(y-\theta)e^{\lambda(y-\theta)}d\theta}{\int f^*(y-\theta)e^{\lambda(y-\theta)}d\theta} \\ &= \frac{e^{-\frac{1}{2\sigma^2}(y-\theta)^2+\lambda(y-\theta)}d\theta}{\int e^{-\frac{1}{2\sigma^2}(y-\theta)^2+\lambda(y-\theta)}d\theta} \\ &= \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2\sigma^2}(\theta-(y-\sigma^2\lambda)^2} \\ &= N(y-\sigma^2\lambda,\sigma^2). \end{aligned}$$

This implies that $|y - E(\theta|y)| \rightarrow \sigma^2\lambda$, the result given by Sansó and Pericchi (1992), see also Pericchi, Sansó, & Smith (1993), and Lucas (1993).

Note, because of the symmetry in the problem, we can reverse the role of the prior and likelihood. For example, if the likelihood is heavier, we see that the marginal tends to the pivot density, and the posterior to the prior, with the appropriate correction factors.

The main power of these results is that once we know which classes the distributions belong to, we immediately see how our estimate for θ will behave as $y \rightarrow \infty$. While these are asymptotic results, we have found they provide reasonable approximations for intermediate data values. Moreover, we can extend the analysis to determine general behavior for additional levels in the hierarchy, scale parameters, and multidimensional parameters.

3.1 Higher Levels

Many applications of hierarchical models contain three or more levels. For example, in educational research, we may have repeat observations on students that are grouped by classroom. Thus, we can see how both the student level and classroom level parameter estimates will behave by repeatedly applying the above theorem.

In general, in a three level model, the behavior of the first level parameter will depend on the first level distribution and the convolution of the higher level densities. Thus, if either of the higher level densities are Heavy, while the first density is not, the information from all higher levels will be asymptotically ignored in the posterior density. If only the first level is Heavy, the posterior tends to the heavier of the remaining densities. Similarly, when the second level parameter is a location parameter, we can describe the behavior of its posterior density. In this case, if only the first level is Heavy, the posterior tends to the prior, with mean at the third level mean. Clearly, the value specified for the third level parameter becomes important. This analysis can easily be extended to additional levels.

3.2 Scale Parameters

Theorem 1 assumes θ is a location parameter, however, in many applications it would be fruitful to know the sensitivity of the variance component estimates. Often, we can obtain similar theory for scale parameters by reparameterizing them as location parameters. This theory applies when the likelihood, $l(\sigma|\mathbf{y})$, is proportional to $l(\frac{s(\mathbf{y})}{\sigma})$. A likelihood of this form can be reexpressed as $l(\frac{s(\mathbf{y})}{\sigma}) = g(\log s(\mathbf{y}) - \log\sigma)$. Transforming to $\theta = \log\sigma$, θ is a location parameter and when Theorem 1 is applicable, we have an approximation for the posterior of θ as $\log s(\mathbf{y}) \rightarrow \infty$. While the NLRs of the transformed densities are often not regularly oscillating, in practice we have found the resulting approximations to still be quite reasonable. We are currently attempting to generalize the behavior when location and scale parameters are assumed unknown simultaneously.

3.3 Multidimensional Problems

When Y is a p variate random vector with mean $\theta = (\theta_1, \ldots, \theta_p)$, and we assume the y_i's are conditionally independent and the θ_i's are exchangeable, we can apply Theorem 1 in each coordinate. Thus, if one coordinate, say y_j becomes large, the limiting behavior for the posterior mean for θ_j will converge to some limiting value as dictated by Theorem 1. When the variance components are assumed known, the posterior means for the other coordinates will behave as if the outlying coordinate did not exist, displaying the expected shrinkage phenomena. When the variance components are modeled as unknown, the mean components will still be linked together, and thus the other components will not completely reject the outlier. This was shown to be true for the Normal-Cauchy case by Angers and Berger (1991).

4 IMPLEMENTATION

Since class membership of the tails determines model behavior, if we can obtain estimates for a representative from each class, we will know how all members of that class behave as $y \rightarrow \infty$. Below we describe a Gibbs Sampling implementation using scale mixtures of Normals that allows estimation for Normal (Very Light), Double Exponential (Light) and t densities (Med-Heavy) by multiplying the Normal scale parameter by λ and then specifying a density for λ. If we wanted to model a particular density that is not easily implemented, we may alternatively obtain estimates from these prototypes that will be robust to outliers. The final estimates for λ also provide a diagnostic for outliers, see for example, Seltzer (1993) and Racine-Poon (1992).

4.1 Student's t Density

A multivariate t distribution can be obtained by mixing a multivariate Normal distribution with a Gamma. Let $Y \sim N_n(\mu, \lambda\Sigma)$ and $\lambda \sim IG(\frac{\nu}{2}, \frac{\nu}{2})$. Then $Y \sim t_\nu(\mu, \Sigma)$ once we integrate out λ because

$$p(y) = \int_0^\infty N_n(\mu, \lambda\Sigma) IG(\nu/2, \nu/2) d\lambda$$

$$= \frac{\nu^{\nu/2}\Gamma(\frac{n+\nu}{2})}{|\Sigma|^{1/2}\pi^{n/2}\Gamma(\frac{\nu}{2})} \left[\nu + (y-\mu)'\Sigma^{-1}(y-\mu)\right]^{-(\frac{n+\nu}{2})}.$$

The conditional distribution for $\lambda|y$ will then be Inverse Gamma($\frac{\nu+n}{2}, \frac{1}{2}s^2 + \nu$), and we can immediately add this distribution to a Gibbs Sampler.

4.2 Double Exponential/Laplace Density

In the univariate case, the Double Exponential can be found by mixing a Normal with an Exponential with mean 2 (Andrews & Mallows, 1974):

$$\frac{1}{2\sigma}e^{-\frac{|y-\mu|}{\sigma}} = \int_0^\infty \frac{1}{\sigma\sqrt{2\pi\lambda}} e^{-\frac{1}{2\lambda\sigma^2}(y-\mu)^2} \frac{1}{2} e^{-\frac{\lambda}{2}} d\lambda$$

Note, this density has variance $2\sigma^2$, so we often use $\frac{\sigma^2}{2}$ as the variance of the Normal distribution. If we extend this idea to the multivariate case, the resulting multivariate density for y unconditional on λ is the *symmetric multivariate Bessel distribution.* To see this, let $Y \sim N_n(\mu, \lambda\frac{\sigma^2}{2}\Sigma)$ and $\lambda \sim Exp(2)$. Then

$$\begin{aligned} f(y) &= \frac{1}{2}\pi^{-\frac{n}{2}}\sigma^{-n}\int_0^\infty \lambda^{-\frac{n}{2}} e^{-\frac{1}{2}[\frac{1}{\lambda}s^2+\lambda]} d\lambda \\ &= 2\frac{1}{2}\pi^{-\frac{n}{2}}\sigma^{-n}s^{-\frac{n}{2}}\frac{1}{2}\int_0^\infty u^{-\frac{n}{2}} e^{-\frac{1}{2}s[\frac{1}{u}+u]} du \\ &= \pi^{-\frac{n}{2}}\sigma^{-n}s^{1-\frac{n}{2}}K_{1-\frac{n}{2}}(s) \end{aligned}$$

where $s^2 = \frac{(Y-\mu)'\Sigma^{-1}(Y-\mu)}{\sigma^2/2}$, and $K_\xi(w)$ is the modified Bessel function of the third kind. The final equation is the Multivariate Bessel density (Fang, Kotz, & Ng, 1990) with parameters $a = 1 - \frac{n}{2}$ and $b = \frac{\sigma}{\sqrt{2}}$. The density is elliptically symmetric. If $n = 1$ the density is the univariate Double Exponential. If $n = 2$, the distribution has been labeled the *bivariate Laplace* distribution. Thus, this mixture provides us with a variant of the multivariate Double Exponential, and we refer to the density resulting from this mixture as a Double Exponential. The full conditional for $\lambda|y$ will be Generalized Inverse Gaussian($1 - \frac{n}{2}, 1, s^2$).

4.3 Gibbs Implementation

By adding λ to our hierarchy, the distributions conditional on λ will be Normal and we can utilize conjugacy to find their exact form. The Inverse Gamma is generated by inverting a Gamma random variable. To generate from the Generalized Inverse Gaussian, GIG, we use a rejection algorithm. If $\gamma < 0$ we take the reciprocal of a GIG($-\gamma, \alpha, \beta$). If $\gamma > 1$ the density is log concave and we use the "non-universal rejection algorithm" given by Devroye (1986). Let $f(x)$ represent our GIG density function, and set $h(x) = \log f(x)$. Since f is log-concave, h can be majorized by the derivative of h at any point, which corresponds to fitting an exponential curve over f. Thus, we use a piecewise majorizing function, $g(x)$, for $f(x)$, where the first piece is an exponential curve, the second piece is f evaluated at the mode, and the third piece is another exponential curve. We select points a and b to attach the exponentials so that the area under $g(x)$ is minimized. Let $m = \frac{(\gamma-1)+\sqrt{(\gamma-1)^2+\alpha\beta}}{\alpha}$ be the mode of f, f_l the tail to the left of the mode, and f_r the right tail. Theorem 2.6 of Devroye states that the area will be minimal if we choose a and b such that

$$m + a = f_r^{-1}(\frac{f(m)}{e})$$
$$m - b = f_l^{-1}(\frac{f(m)}{e}).$$

We use a binary search to find these cross points. This tells us where to attach the exponential curves and gives us a piecewise majorizing function.

If $\gamma < 1$, we use the above algorithm for the three regions to the left of the infection point, $a_i = \frac{\alpha}{1-\gamma}$, and majorize the region to the right of the inflection point by a pareto curve ($\frac{a_i}{x^2}$). In this case we need to calculate the Bessel function in the constant of integration in the GIG density. The inflection is always to the right of the mode. If the inflection point falls to the left of $m+a$, then we attached the second exponential curve at the inflection point. Figure 1 shows the density and the piecewise majorizing function for $\alpha = 1, \gamma = .5$. If $\gamma = .5$ we actually generate from a Reciprocal Inverse Gaussian by inverting an observation from an Inverse Gaussian. To sample from an Inverse Gaussian we use the algorithm of Michael, Schucany, and Hass given in Devroye (1986) using a many-to-one transformation (p. 148-149). In our hierarchical models, $\gamma = 1 - \frac{n}{2}$ so the only val-

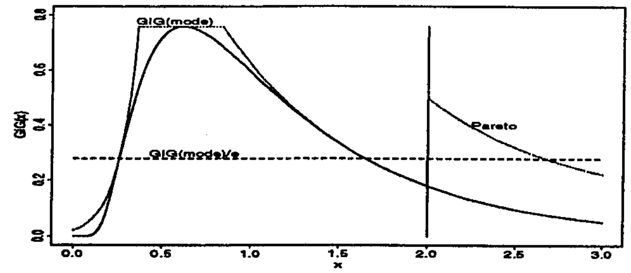


Figure 1: Generating from a GIG density

ues of $0 \leq \gamma < 1$ we need to consider are $\gamma = 0, .5$ as n is an integer. The above algorithm extends that used by Carlin and Polson (1991) which dealt with the case $n = 1$.

5 CONCLUSION

In summary, the negative log rate provides a characterization of densities based on their tail behavior. This characteristic determines when a Bayes estimate compromises or rejects sources of information. This knowledge aids in model selection, by knowing the consequences of our assumptions in the presence of outlying data. Given a particular density selection, the theory also indicates when we could substitute alternative distributions that have similar behavior but may be more tractable. The Gibbs Sampling implementation allows estimation for a prototype from each class.

6 REFERENCES

ANDREWS, D.R. & MALLOWS, C.L. (1974). "Scale Mixtures of Distributions". Journal of the Royal Statistical Society, Series B. 36, p. 99-102.

ANGERS, J.-F. & BERGER, J.O. (1991). "Robust Hierarchical Bayes Estimation of Exchangeable Means". Canadian Journal of Statistics. 19, p. 39-56.

BERMAN, S.M. (1992) "The Tail of the Convolution of Densities and its Application to a Model of HIV-Latency Time". The Annals of Applied Probability. 2(2), p. 481-502.

BROWN, L.D. (1988). "The Differential Inequality of a Statistical Estimation Problem", in **Statistical Decision Theory and Related Topics IV**, Vol. 1. S. S. Gupta, J. O. Berger, Eds. Berlin: Springer Verlag. p. 299-324.

CARLIN, B.P. & POLSON, N.G. (1991). "Inference for Nonconjugate Bayesian Models Using the Gibbs Sampler". The Canadian Journal of Statistics. 19(4), p. 399-405.

CHANCE, B.L. (1994). "Characterizing Behavior and Estimation for General Hierarchical Multivariate Linear Regression Models". Ph.D. Thesis, Cornell University.

DAWID, A.P. (1973) "Posterior Expectations for Large Observations". Biometrika, 60, p. 664-667.

DEVROYE, L. (1985). **Non-Uniform Random Variate Generation.** New York: Springer-Verlag.

FANG, K-T., KOTZ, S., & NG, K.W. (1990). "Symmetric Multivariate and Related Distributions". London: Chapman & Hall.

GOMEZ-VILLEGAS, M.A. & MAIN, P. (1992) "The Influence of Prior and Likelihood Tail Behavior on the Posterior Distribution", in **Bayesian Statistics 4**. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, eds. Oxford: Oxford University Press. p. 661-667.

HILL, B.M. (1974) "On Coherence, Inadmissibility and Inference About Many Parameters in the Theory of Least Squares", in **Studies in Bayesian Econometrics and Statistics**. S.E. Fienberg and A. Zellner, eds. Amsterdam: North-Holland Pub. Co. p. 555-584.

LINDLEY, D.V. & SMITH, A.F.M. (1972). "Bayes Estimates for the Linear Model". Journal of the Royal Statistical Society, Series B. 34, p. 1-41.

LUCAS, T.W. (1993). "When is Conflict Normal?". JASA. 88(424), p. 1433-1437.

MEEDEN, G., & ISAACSON, D. (1977) "Approximate Behavior of the Posterior Distribution for a Large Observation", The Annals of Statistics, 5(5), p. 899-908.

O'HAGAN, A. (1988) "Modeling with Heavy Tails", in **Bayesian Statistics 3**. J.M. Bernardo, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, eds. Oxford: Oxford University Press, p. 345-359.

PERICCHI, L.R., SANSÓ, B., SMITH, A.F.M. (1993). "Posterior Cumulant Relationships in Bayesian Inference Involving the Exponential Family". JASA. 88(424), p. 1419-1426.

RACINE-POON, A. (1992). "SAGA: Sample Assisted Graphical Analysis", in **Bayesian Statistics 4**. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, eds. Oxford: Oxford University Press. p. 389-404.

SANSÓ, B. & PERICCHI, L.R. (1992). "Near Ignorance Classes of Log-Concave Priors for the Location Model", Test. 1(1), p. 39-46.

SELTZER, M.H. (1993). "Sensitivity Analysis for Fixed Effects in the Hierarchical Model: A Gibbs Sampling Approach". Journal of Educational Statistics. 18(3), p. 207-235.

PREDICTING URBAN OZONE LEVELS AND TRENDS WITH SEMIPARAMETRIC MODELING

Feng Gao[1], Jerome Sacks[2] and William J. Welch[3]

NISS, P.O. Box 14162, Research Triangle Park, NC 27709-4162

1 Introduction

High ozone concentration in the troposphere is believed to be harmful to human health and to crops (see National Research Council (1991)). The surface ozone concentration level is affected by the strengths of sources and precursor emissions, and by meteorological condition. To assess that part of the trend in ozone concentration levels that cannot be accounted for by meteorology, we need to build models which relate ozone to meteorology.

In Bloomfield, Royle and Yang (1993), nonlinear least squares methods were used to model the dependence of ozone on meteorology, and to estimate the trends. That report focuses on the urban Chicago area.

In this report, a semiparametric modeling technique is used to build models that relate ozone to meteorology.

2 Semiparametric Model

The ozone concentration value to be modeled here is the daily network *typical value*. To obtain the daily network typical value, the least absolute deviations decomposition (or the median polish decomposition, see Tukey (1977)) of $y_{d,s}$, the maximum concentration on day d at station s, was performed for all the 45 ozone monitoring stations in the urban Chicago area:

$$y_{d,s} = \mu' + \alpha'_d + \beta'_s + \epsilon'_{d,s}$$

The daily network typical value is then defined as $\mu' + \alpha'_d$. The decomposition was also used to impute the missing data. This daily network typical value is called the network average in Bloomfield et al. (1993). The unit for ozone concentration is parts per billion (ppb).

The same meteorological variables adopted by Bloomfield et al. (1993) are used here. The surface weather data were taken from O'Hare Airport and the upper air weather data were taken from a station at Peoria in the same period ozone data were taken. The variables used are:

- maximum temperature from 9:00 am to 6:00 pm (maxt)
- 12 noon wind speed (wspd)
- 24 hr ave. wind vector (meanu and meanv)
- 12 noon relative humidity (rh)
- 12 noon visibility (vis)
- 12 noon opaque cloud cover (opcov)
- 7 am wind speed at 700 mb (wspd700)
- 24 hr ave. temp. lagged 1 and 2 days (tlag1 and tlag2)
- 24 hr ave. wind speed lagged 1 day (wlag)
- 24 hr ave. relative humidity lagged 1 day (rhlag)

Also used is a variable for year, which takes the integer values 1,2,..., 11, corresponding to years 1981 - 1991, and a variable for day taking values from 1 to 365 to reflect seasonal effects.

On day i, in year j, with meteorological condition met, where met is a 12-dimensional vector of the above meteorological variables, let $x = (met, i, j)$. So x is a 14-dimensional

[1] National Institute of Statistical Sciences (NISS), P. O. Box 14162, Research Triangle Park, NC 27709-4162. Research supported in part by the U.S. Environmental Protection Agency under Cooperative Agreement #CR819638-01-0 and National Science Foundation Grant DMS-9208758.

[2] National Institute of Statistical Sciences (NISS), P. O. Box 14162, Research Triangle Park, NC 27709-4162. Research supported in part by the U.S. Environmental Protection Agency under Cooperative Agreement #CR819638-01-0.

[3] Department of Statistics and Actuarial Science, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada. Research supported in part by the U.S. Environmental Protection Agency under Cooperative Agreement #CR819638-01-0.

vector $x = (\xi_1, \ldots, \xi_{14})$. The response $y(x)$ (the network typical value) is assumed to be a realization of a stochastic process, $Y(x)$:

$$Y(met,i,j) = \beta_j + Z(met,i,j) + \varepsilon_{ij} \qquad (1)$$

where β_j are constants, $j = 1, 2, \ldots, 11$, $Z(x) = Z(met,i,j)$ is a zero mean Gaussian process with covariance function $\mathrm{Cov}(Z(x), Z(x')) = \sigma_Z^2 R(x,x')$, and $\varepsilon_{ij} \sim \mathcal{N}(0, \sigma_\varepsilon^2 I)$. See Sacks, Welch, Mitchell and Wynn (1989) for more discussion.

Assume, as in Sacks et al. (1989), that the covariance between $Z(x)$ and $Z(x')$ is

$$\sigma_Z^2 R(x,x') = \sigma_Z^2 \exp(-\sum_{k=1}^{14} \theta_k |\xi_k - \xi_k'|^{p_k}) \qquad (2)$$

where $x = (\xi_1, \ldots, \xi_{14})$, $x' = (\xi_1', \ldots, \xi_{14}')$, $\theta_k \geq 0$, $1 \leq p_k \leq 2$, $k = 1, \ldots, 14$. This class of stationary processes provides us with a wide range of functions.

Given the data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ for q consecutive years starting from year 1 (1981) with n_j data points in year j and $n_1 + \cdots + n_q = n$ and, provided σ_Z, σ_ε and $R(\cdot,\cdot)$ are known, the best linear unbiased predictor (BLUP) $\hat{y}(x)$ at a new point x in year j can be written as (see Sacks et al. (1989))

$$\hat{y}(x) = \hat{\beta}_j + \hat{Z}(x) = \hat{\beta}_j + r'(x)C^{-1}(y - F\hat{\beta}) \qquad (3)$$

where $y = (y_1, y_2, \ldots, y_n)$, $C = \mathrm{Corr}(y) = (\sigma_Z^2/\sigma^2)R + (\sigma_\varepsilon^2/\sigma^2)I$, where $\sigma^2 = \sigma_Z^2 + \sigma_\varepsilon^2$, and $R = \{R(x_i, x_j), 1 \leq i \leq n; 1 \leq j \leq n\}$, the $n \times n$ matrix of correlations among Z's at the data points, $r(x) = (\sigma_Z^2/\sigma^2)[R(x_1,x), \ldots, R(x_n,x)]'$,

$$F = \begin{pmatrix} \vec{1}_{n_1\times 1} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{1}_{n_2\times 1} & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \cdots & \vec{1}_{n_q\times 1} \end{pmatrix}_{n\times q}$$

and $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_q)' = (F'C^{-1}F)^{-1}F'C^{-1}y$, which is the usual generalized least-squares estimate of $\beta = (\beta_1, \ldots, \beta_q)'$.

In the model, values of p indicate smoothness of the response surface as a function of the corresponding variables. Larger values of θ usually indicate greater importance of the corresponding variables if the variables are on normalized scales.

To obtain the unknown parameters σ_Z, σ_ε, θ's and p's, maximum likelihood estimate (MLE) method is used. These estimates are then used in (3) to predict the response surface.

This work focuses on the period from May 15 to Sept. 15, the period when ozone concentration is high. This period is divided into 4 smaller periods: May 15 - June 15, June 15 - July 15, July 15 - Aug. 15 and Aug. 15 - Sept. 15. The reasons are: First, the assumption of stationarity of Z within a shorter time period is more plausible. Secondly, fitting a model for each of the 4 periods separately reduces the computational burden. For more details, please see Gao, Sacks and Welch (1994).

3 Modeling the Network Typical Value

A model is fitted using data from 1981 to 1991 for each of the 4 periods.

3.1 Important Variables

To see which meteorological variables have strong effects, we rescale them so that each meteorological variable ranges over [0,1]. The MLEs of the θ's and p's with the rescaled meteorological variables and the rescaled variables day and year are given in Table 1.

The estimated θ for year was 0 for the first 3 monthly periods. For the fourth monthly period, the estimated θ for year was small, indicating that year was not an important variable. Because the adjusted trend of ozone could be unambiguously interpreted through the β_j's if θ for year was 0 (see Section 3.3), we choose to set the θ for year equal to 0 in the fourth period as well.

From the table, it can be seen that temperature, relative humidity and wind (through wspd, wlag, meanu, meanv and/or wspd700) are consistently important across the months. For more discussion, please see Gao et al. (1994).

3.2 Quality of the Fitted Models

To check the quality of the model fit, the *cross validation root mean square error* (CVRMSE) was calculated. If the model fit is good, the CVRMSE should be close to σ_ε or its MLE.

Table 2 lists the MLEs of σ_Z and σ_ε and the CVRMSEs for the fitted models. The table shows that the model fits are generally good. The values of CVRMSE are close to the values of root mean square residual from the parametric model fitting in Table 6 of Bloomfield et al. (1993). For more discussion, please see Gao et al. (1994).

3.3 Trend Estimation

It is possible to interpret the adjusted trend through the β_j's in the model when the variable year does not appear in the stochastic process part of the model $Z(\cdot)$, or equivalently when θ for year is 0. Under this circumstances, if *met* is held fixed, the change from year to year is, except for random errors ε, reflected in the differences of the β_j's. Therefore the adjusted trend is defined as the trend in the β_j's.

Let $\beta_j^* = \hat{\beta}_j + (\bar{y} - \bar{\hat{\beta}})$, then $\bar{\beta}^* = \bar{y}$. These β_j^*'s can be interpreted as the adjusted (for meteorology) averages of ozone level across the years while the simple yearly averages $\bar{y}_j$'s are the unadjusted averages. The time series plots in Figure 1 demonstrate that a large portion of the variability in the unadjusted averages is eliminated in the adjusted averages. This portion of the variability is caused by meteorology. The plots suggest a linear trend for the adjusted averages. The lines in the plots are the least square regression lines. Let $\hat{a}$ be the intercept at year= 81 and $\hat{b}$ be the slope of the line, then the estimate of the adjusted trend is

$$\widehat{trend} = 10 \times \frac{\hat{b}}{\hat{a}} \qquad (\%/\text{decade}). \tag{4}$$

Based on the model and using MLEs of the parameters, the standard errors of the estimates of the trend can be estimated. The standard errors of the estimates of the trend can also be estimated by jackknifing by day (see Chapter 8 of Mosteller and Tukey (1977)). Also see Gao et al. (1994) for more details. The estimates of the trends and their standard errors are listed in Table 3.

3.4 Predictions

The models constructed can be used to predict behavior of ozone in future years as a function of meteorology. The results in Gao et al. (1994) show that the model predictions closely match the actual ozone levels.

4 Conclusions

The semiparametric modeling technique is shown to provide a good way to model the ozone concentration as a function of meteorology. This method can be used to assess the adjusted trends. The models can also be used to predict ozone levels from meteorology.

It is found that for the urban Chicago area, there are significant downward trends for the network typical ozone values after adjusting for meteorology for the periods June 15 - July 15 and Aug. 15 - Sept. 15 over the 11 years studied (see Table 3).

In Bloomfield et al. (1993), for the period of Apr. 1 - Oct. 31, the adjusted trend for the network typical values is found to be −2.7%/decade with a (jackknife) standard error of 3.4%/decade. Results from the two reports appear to be consistent.

References

Bloomfield, P., Royle, A. and Yang, Q. (1993). Accounting for Meteorological Effects in Measuring Urban Ozone Levels and Trends, *Technical Report No. 1.* National Institute of Statistical Sciences, P.O.Box 14162, Research Triangle Park, NC 27709-4162.

Gao, F., Sacks, J. and Welch, W. (1994). Predicting the Urban Ozone Levels and Trends with Semiparametric Modeling, *Technical Report No. 14.* National Institute of Statistical Sciences, P.O.Box 14162, Research Triangle Park, NC 27709-4162.

Mosteller, F. and Tukey, J. W. (1977). *Data Analysis and Regression*, Addison-Wesley, Reading, Massachussetts.

National Research Council (1991). *Rethinking the Ozone Problem in Urban and Regional Air Pollution*, National Academy Press, Washington, D.C.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989). Design and Analysis of Computer Experiments, *Statistical Science* **4**: 409–435.

Tukey, J. W. (1977). *Exploratory Data Analysis*, Addison-Wesley, Reading, Massachussetts.

Table 1: Estimates of θ's and p's for models for the network typical value with meteorological variables rescaled.

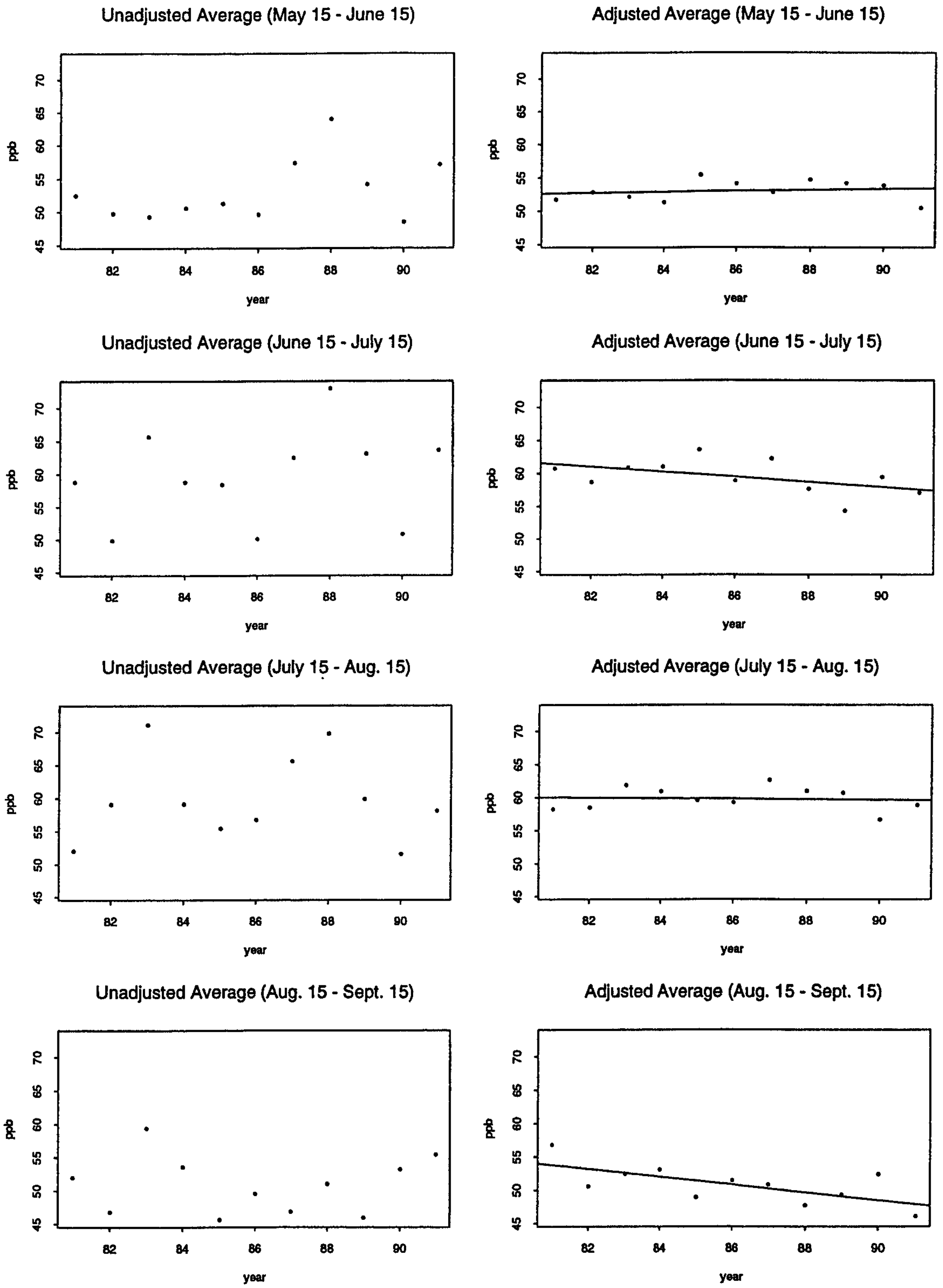
	May 15 - June 15		June 15 - July 15		July 15 - Aug. 15		Aug. 15 - Sept. 15	
variables	θ	p	θ	p	θ	p	θ	p
maxt	3.3222	2	4.8540	2	0.8202	2	1.7082	2
wspd	0.2368	2	0.0000	2	0.3261	2	0.0000	2
meanu	0.0000	2	0.7052	1.185	0.0733	1.435	0.3849	1
meanv	0.0000	2	1.4913	2	0.5097	2	1.6454	2
rh	1.7655	2	0.7756	2	0.5506	2	1.9768	2
vis	0.1833	2	1.1844	2	0.0333	2	0.6480	2
opcov	0.0000	2	0.0731	2	0.0000	2	0.0617	2
wspd700	0.5513	2	0.0000	2	0.0170	2	0.1922	2
tlag1	0.0000	2	0.5948	2	0.0000	2	0.0000	2
tlag2	0.0928	2	0.0000	2	0.0462	2	0.2011	2
wlag	0.0481	1	0.6122	2	0.0000	2	2.2078	2
rhlag	0.0000	2	0.3031	2	0.0000	2	0.1955	2
day	0.0939	2	0.1270	2	0.0000	2	0.1172	2
year	0.0000	2	0.0000	2	0.0000	2	0.0000	2

Table 2: Estimates of σ_Z and σ_ε, and CVRMSE for models for the network typical value.

Models	$\hat{\sigma}_Z$	$\hat{\sigma}_\varepsilon$	CVRMSE
May 15 - June 15	17.051	7.028	7.606
June 15 - July 15	17.584	7.028	8.433
July 15 - Aug. 15	33.194	9.236	9.828
Aug. 15 - Sept. 15	15.263	6.281	7.680

Table 3: Estimates of trends and their standard errors for the adjusted averages for the network typical value.

	Model Estimates			Jackknifed Estimates		
Models	Trend	Standard Error	t Value	Trend	Standard Error	t Value
May 15 - June 15	0.0139	0.0280	0.4964	0.0149	0.0288	0.5174
June 15 - July 15	−0.0635	0.0285	2.2281	−0.0651	0.0287	2.2683
July 15 - Aug. 15	−0.0074	0.0313	0.2364	−0.0093	0.0310	0.3000
Aug. 15 - Sept. 15	−0.1094	0.0330	3.3152	−0.1146	0.0290	3.9517



Unadjusted Average (May 15 - June 15)

Adjusted Average (May 15 - June 15)

Unadjusted Average (June 15 - July 15)

Adjusted Average (June 15 - July 15)

Unadjusted Average (July 15 - Aug. 15)

Adjusted Average (July 15 - Aug. 15)

Unadjusted Average (Aug. 15 - Sept. 15)

Adjusted Average (Aug. 15 - Sept. 15)

Figure 1: Adjusted and unadjusted averages of the network typical values.

AUTHOR INDEX

LIST OF PARTICIPANTS

Edgar Acuna
University Puerto Rico-Mayaeuez
Department of Mathematics
Mayaeuez Puerto Rica 00680
e_acuna@upr1.upr.clu.edu

David M. Allen
University of Kentucky
Department of Statistics
Lexington, KY 40506
allen@ms.uky.edu

Stuart Altschuler
Merrill Lynch
P.O. Box 9065
Princeton, NJ 08543

Andy Andrews
University of Michigan
School of Business Administration
Ann Arbor, MI 48109-1234
andy_andrews@um

David, Andrews
Rice University
1615 South Blvd
Ann Arbor, MI 48104

Tim Arnold
North Carolina State University
Statistics Department Box 8203
Raleigh, NC 27695-8203
arnold@stat.ncsu.edu

Leonardo Auslender
AT&T
8 Sagamore Ave
Edison, NJ 08820
attmail!dmsmodel!leonardo

Stan Azen
Univ of Southern California
Department of Preventive Medicine
1420 San Pablo Street PMB B101
Los Angeles, CA 90033

Keith A. Baggerly
Rice University
Department of Statistics
P.O. Box 1892
Houston, TX 77030
kabagg@stat.rice.edu

John Bailer
Miami University/NIOSH
Department of Math & Statistics
Oxford, OH 45056
ajbailer@miavx1.muohio.edu

Barbara Bailey
North Carolina State University
P.O. Box 8203
Raleigh, NC 27695

Alfred Balch
Bristol-Myers Squibb
Pharmaceutical Research Institute
5 Research Parkway
Wallingford, CT 06492
balch@bms.edu

Huiman X Barnhart
Emory Univ Schl of Public Health
Division of Biostatistics
1599 Clifton Road NE
Atlanta, GA 30329
hxb@panda.sph.emory.edu

Andrew R Barron
Yale University
Department of Statistics
New Haven, CT 06520
barron@stat.yale.edu

Edward Barrows
ManTech Environmental
2 Triangle Drive
Research Triangle Park, NC 27709

Cathleen Barrows
North Carolina State University
112 Willoughby Lane
Cary, NC 27513
Barrows@STAT.NCSU.edu

Bruce Belanger
Becton Dickinson Rsrch Cntr
P.O. Box 12016
Research Triangle Park, NC 27709
bab@bdrc.bd.edu

Kerry G Bemis
Eli Lilly and Company
Lilly Corporate Center
Indianapolis, IN 46285
Kristin P. Bennett
Rennsselaer Polytechnic Inst
Mathematical Sciences Department
Troy, NY 12180-3590
bennek@rpi.edu

Lance Benson
Klemm Analysis Group Inc
1785 Massachusetts Ave NW
Washington, DC 20036

Al Best
Virginia Commonwealth Univ
Box 980032
Richmond, VA 23005
best@gems.vcu.edu

Lynne Billard
University of Georgia
Department of Statistics
204 Statistics Building
Athens, GA 30602-1952
lynne@marie.stat.uga.edu

Tom Birkett
USDA/NASS
Washington, D.C. 20009

Jean-Louis Blanchard
Electricite de France
5 Residence la Fontaine
91480 Quincy sans Senart France
Jean-Louis.Blanchard@der.edf.Fr

Mary Ellen Bock
Purdue University
Statistics Department
1399 Math Science Building
W. Lafayette, IN 47907
mbock@stat.purdue.edu

Dennis Boos
North Carolina State Univ
Department of Statistics
Box 8203
Raleigh, NC 27695-8203
boos@stat/ncsu.edu

Jason Brown
University of Missouri-Columbia
Department of Statistics
222 Math Sciences Building
Columbia, MO 65211
Brown@Stat.Missouri.edu

Don Brown
University of Virginia
Institute for Parallel Computation and
Department of Systems Engineering
Charlottesville, VA 22901-2442

Robert Brown
Statistical Consultant
2637 Anthony Dr
Colmar, PA 18915

Steve Bryant
Ntn'l Cntr Biotec Info NLM/NIH
National Institutes of Health
8600 Rockville Pike
Bethesda, MD 20894
bryant@ncbi.nlm.nih.gov

Wray L. Buntine
RIACS/NASA Ames Rsrch Cntr
Mail Stop 269-2
Moffet Field, CA 94035-1000
wray@kronos.arc.nasa.gov

Thomas E. Burk
University of Minnesota
Department of Forest Resources
115 Green Hall, 1530 N Clevland Ave
St. Paul, MN 55108
teb@dendron.forestry.umn .edu

Jan W. Buzydlowski
American College of Radiology
1101 Market Street
14th Floor
Philadelphia, PA 19107
JBUZYDLOWSKI@ACR.ORG

Jackie Callaghan
George Mason University
7610 Glenolden Place
Manassas, VA 22111
JCALLAGH@mason1.gmu.edu

Angelo Canty
University of Toronto
Department of Statistics
24 Southport St. #455
Toronto, Ontario M6S 4Z1 Canada
angelo@utstat.utoronto.ca

Vincent Carey
Harvard Medical School
Channing Laboratory
180 Longwood Ave
Boston, MA 02115
stvjc@gauss.med.harvard.edu

Daniel Carr
George Mason University
Center for Computational Statistics
157 Science-Technology Building #2
Fairfax, VA 22030
dcarr@galaxy.gmu.edu

Philippe Castagliola
Ecole des Mines de Nantes
3 rue Marcel Sembat
44040 NANTES
Cedex 04 France
pcasta@auto.emn.Fr

Beth Chance
Cornell University
Statistics Center
228 ETC, 2nd Floor
230 Bryant Ave
Ithaca, NY 14850
nyerges@orie.cornell.edu

Chung-Kuei Chang
Rhone - Poulenc Rorer
500 Arcola Road
Collegeville, PA 19426

Guang-hwa Chang
Youngstown State University
Department of Mathematics
Youngstown, OH 44555
chang@math.ysu.edu

Victoria Chen
Georgia Institute of Technology
School of Industrial & Systems Engineering
Atlanta, GA 30332-0205
vchen@isye.gatech.edu

Tar Timothy Chen
NIH-NCI
Bethesda, MD 20892
tchen@helix.nih.gov

Shaohsin Chen
Kansas State University
1823 Platt Street
Manhattan, KS 66502
shaohsin@cecil.stat.ksu.edu

Yang Chen
Duke University
311 S. Lasalle 45-D
Durham, NC 27705
yang@isds.duke.edu

Guang Chen
Kansas State University
Department of Statistics
Manhattan, KS 66502
guang@cecil.stat.ksu.edu

Cheng Cheng
Upjohn Laboratories
301 Henrietta Street, 7247-267-1
Kalamazoo, MI 49007
ccheng0@intnet.upj.edu

Hugh Chipman
University of Waterloo
Department of Statistics, Univ. of Michigan
1444 Mason Hall, 419 S. State Street
Ann Arbor, MI 48109-1027
hachipma@stat.lsa.umich.edu

Gary Churchill
Cornell University
Plant Breeding and Biometry Department
337 Warren Hall
Ithaca, NY 14853
gary@amanita.bio.cornell.edu

Linda Clark
AT&T Bell Labs
Room 2C-273
600 Mountain Avenue
Murray Hill, NJ 07974
lac@research.att.edu

Dianne Cook
Iowa State University
Department of Statistics , ISU
323 Snedecor Hall
Ames, IA 50011
dicook@iastat.edu

Bill Cox
USEPA MD-14
Research Triangle Park, NC 27711

Lawrence Cox
USEPA AREAL MD 75
Research Triangle Park, NC 27711
cox.larry@epamail.epa.gov

John P Creason
USEPA MD-55
Research Triangle Park, NC 27711

Kathleen Cronin
Cornell University
Statistics Center
228 ETC 230 Bryant Ave
Ithaca, NY 14853
cronin@orie.cornell.edu

Rainer Dahlhaus
Universitat Heidelberg
Institut fur Angewandte Mathematik
Im Neuenheimer Feld 294
Heidelberg, Germany
dahlhaus@statlab.uni-heidelberg.de

James Daughtery
Princpia Information Group
3725 National Dr Suite 230
Raleigh , NC 27612

Lorraine Denby
AT&T Bell Labs
Room 2C-255
600 Mountain Avenue
Murray Hill, NJ 07922
ld@research.att.edu

Lih-Yuan Deng
Memphis State University
Department of Mathematical Sciences
Memphis, TN 38152
DENGL@HERMES.MSCI.MEMST.edu

Thomas F Devlin
Montclair State University
Math & Computer Science Department
10 Symor Dr
Convent Station, NJ 07960-6526
devlin@mozart.Montclair.edu

Sudha Dhandapani
Decision Focus Incorporated
650 Castro Street, Suite 300
Mountain View, CA 94041-2055
sudha@dfi.edu

Valentina Di Francesco
NIH-DCRT-Anlytcl Biostat SEC
9000 Rockville Pike.
Bldg 12 A Room 2039
Bethesda, MD 20850
valeo1F@helix.nih.gov

D.A. Dickey
North Carolina State University
Department of Statistics
Box 8203
Raleigh, NC 27695-8203
dickey@stat.ncsu.edu

Kim-Anh Do
Queensland Univ of Tech
School of Mathematics
Brisbane 4074 Australia
k.do@qut.edu.au

Alan Dorfman
Bureau of Labor
Department of Statistics, Room 4925
2 Massachuesetts Avenue, N.E.
Washington, D.C. 20212-0001
dorfmana@ore.psb.bls.gov

William DuMouchel
Columbia University
Division of Biostatistics
600 W. 168th Street
New York, NY 10032
dumouch@cucis.cis.columbia.edu
Rudolf Dutter

University of Technology- Vienna
Department of Statistics & Probability Theory
Wiedner Hauptstr. 8-10
Vienna, Austria 1040
dutt@swtm1.tuwien.ac.at

Alan Eaton
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

William Eddy
Carnegie Mellon University
Statistics Department
Pittsburgh, PA 15213
bill@stat.cmu.edu

Don Edwards
University of South Carolina
Department of Statistics
Columbia, SC 29208
edwards@milo.stat.scarolina.edu

Sam Efromovich
University of New Mexico
Dept Mathematics and Statistics
Albuquerque, NM 87131
EFROM@MOSKVA.UNM.edu

Stephen Eick
AT&T Bell Labs
Room 1G-351
1000 E. Warrenville Road
Naperville, IL 60566
eick@research.att.edu

John F. Elder IV
Rice University
Department Computational and Applied
Mathematics
Houston, TX 77251-1892
elder@rice.edu

Terry Elrod
University of Alberta
Edmonton, Alberta T6G 2R6 Canada
TELROD@GPU.SRX.UALBERTA.CA

Donald Erdman
SAS Institute Inc
SAS Campus Drive
Cary, NC
SASDJE@UNX.SAS.edu

Richard Evans
SUNYA SPH/2 University Pl.
Albany, NY 12203-3399
re0920@thor.albany.edu

Richard Faldowski
University of North Carolina
Department of Statistics
Chapel Hill, NC 27599

Fred Faltin
GE Corporate R&D Center
PO Box 8
Schenectady, NY 12301

Jianqing Fan
University of North Carolina
Statistics Department
322 Phillips Hall, CB #3260
Chapel Hill, NC 27599

Julia Corbin Fauntleroy
Center for Naval Analyses
4401 Ford Avenue
Alexandria, VA 22302-0268
fauntlej@cna.org

Linda R. Ferguson
UCLA Office Academic Computing
405 Hilgard Avenue
5628 Mathematical Sciences Addition
Los Angeles, CA 90024-1557
cusgerf@mvs.oac.ucla.edu

G Stephen Few
N.C. State
113 Collier Pl, #1B
Cary, NC 27513

Nicholas Fisher
CSIRO
Division of Mathematics & Statistics
Locked Bag 17
North Ryde NSW 2113 Australia
nickf@syd.dms.csiro.au

Randall P Fotiu
Michigan State University
Computer Laboratory, Computer Center
East Lansing, MI 48864
fotiu@msu.edu

David Fram
Belmont Research
84 Sherman St
Cambridge, MA 02140
dfram@belmont.edu

Bob Funderlic
North Carolina State University
P.O. Box 8203
Raleigh, NC 27695

Ron Gallant
University of North Carolina
Department of Economics
CB# 3305 6F Gardner Hall
Chapel Hill, NC 27599-3305
ron_gallant@unc.edu

Stuart Gansky
Univ of North Carolina
415 Ridgefield Rd
Chapel Hill, NC 27514
205gansky@zeus@sph.unc.edu

Feng Gao
Feng
National Inst Statistical Sciences
P.O. Box 14162
Research Triangle Park
NC
27709-4162
gao@niss.rti.org

Alan C. Genz
Washington State University
Department of Mathematics
Pullman, Washington 99164-3113
acg@eecs.wsu.edu

Stephen L George
Duke University Medical Center
2024 West Main Street
Suite B101
Durham, NC 27705
sgeorge@ccstat.mc.duke.edu

Alex Georgiev
Albemarle Corporation
Research and Development Department
8000 GSRI Avenue
Baton Rouge, LA 70820
a.georgiev@ieee.org

Kenneth M. Goldberg
Wyeth-Ayerst Research
145-CC P.O. box 8299
Philadelphia, PA 19101

Arnold Goodman
County of Los Angeles
18231 Hillcrest Circle
Villa Park, CA 92667

Pedro Gozalo
Brown University
Department of Economics,
Box B
Providence, RI 02912
pg@pstc3.pste.brown.edu

Yves L. Grize
CIBA-GEIGY
Ciba-Geigy Ltd, R-1009.Z2.05
BASEL
CH 4002
Switzerland
wgry@ciba-geigy.ch

Alan M. Gross
Bellcore
Red Bank, NJ 07701
amg@bellcore.edu

Antonio Gualtierotti
IDHEAP, University of Lausanne
21, Route de la Maladiere
Chavannes-Pres-Renens
CH-1022
Switzerland
antonio.qualtierotti@idheap.unil.ch

George H. Guirguis
George H.
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
sasghg@unx.sas.edu

Perry Haaland
Perry
Becton Dickinson Rsrch Cntr
P.O. Box 12016
Research Triangle Park, NC 27709
pdh@bdrc.bd.com

Dr. Gerald Hajian
Burroughs Welcome
3030 Cornwallis Road
Research Triangle Park, NC 27709

Charles Hallahan
USDA/ERS
4800 Little Falls Rd
Arlington, VA 22207
HALLAHAN@ERS.BITNET

Eric Hallman
Glaxo, Inc
5 Moore Drive
Research Triangle Park, NC 27709
leh26928@usav01.glaxo.edu

David J. Hand.
The Open University
Department of Statistics
Walton Hall
Milton Keynes
MK7 6AA
United Kingdom
d.j.hand@open.ac.uk

Douglas Haney
Becton Dickinson
1510 Vista Club Circle #101
Santa Clara, CA 95054
Doug-Haney@bdis.edu

David Hardesty
University of South Carolina
Department of Statistics
Columbia, SC 29208
Hardesty@milo.math.scarolina.edu

Janis Hardwick
University of Michigan
Statistics Department
1444 Mason Hall
Ann Arbor, MI 48109-1027
JPHARD@UMICH.edu

Eugene Harris
University of Virginia
P.O. Box 710
Madison, VA 22727

Pamela A. Hartford
Battelle Memorial Institute
Statistics and Analysis Systems
505 King Avenue
Columbus, OH 43201-2693

Wolfgang Hartman
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
saswmh@unx.sas.edu

Vic Hasselblad
Duke University
Center for Health Policy
2200 W. Main St., Suite 230
Durham, NC 27705
HASSE002@mc.duke.edu

Trevor Hastie
AT&T Bell Labs
Room 2C-261
600 Mountain Ave
Murray Hill, NJ 07974
trevor@research.att.edu

Leonard Hearne
George Mason University
Department of Statistics
4400 University Drive
Fairfax, VA 22030
lhearne@galaxy.gmu.edu

William D. Heavlin
Advanced Micro Devices
PO Box 3453 , MS 152
Sunnyvale, CA 94088-3453
bill.heavlin@amd.edu

Richard M. Heiberger
Temple University
Department of Statistics
Philadelphia, PA 19122-2585
rmh@astro.ocis.temple.edu

Karl Heiner
SUNY Newpaltz
1739 Athol Road
Schenectady, NY 12308
kwh@world.std.edu

Roelof Helmers
Cntr for Math & Computer Sci
P.O. Box 94079
1090 GB Amsterdam
The Netherlands
helmers@cwi.ni

Tim Hesterberg
Franklin and Marshall College
Mathematics Department
Lancaster, PA 17604-3003
t_hesterberg@fandm.edu

Bernard R. Hill
Coats America
P.O. Box 368
Marion, NC 28752

Joan F. Hilton
Univ of California - San Francisco
Dept of Epidemiology & Biostatistics
San Francisco, CA 94143-0560
joan@biostat.ucsf.edu

John E. Hinkle
University of Kentucky
Department of Statistics
Lexington, KY 40506
jhinkle@ms.uky.edu

Lasse, Holmstrom
University of Helsinki
Rolf Nevanlinna Institute
P.O. Box 26
Helsinki FIN 00014 Finland
LLH@ROLF.HELSINKI.FI

Dennis E. House
USEPA MD-55
Research Triangle Park. NC 27711

Peter Hovey
Airforce Inst Tech AFIT/ENC
WPAFB, OH 45433-7765
phovey@afit.af.mil

Chaun Chieh Hsu
Univ of Alabama-Birmingham
1719 6th Ave South Room #252
Birmingham, AL 35294
J_HSU@CIWAX.CIRC.UAB.edu

Chaun Chieh Hsu
Chaun Chieh
Univ of Alabama-Birmingham
1719 6th Ave South
Room #252
Birmingham, AL 35294
J_HSU@CIWAX.CIRC.UAB.edu

Xin Huang
Univ of Alabama-Birmingham
153 WTI Biostatistic Unit
University Station
Birmingham, AL 35294
huang@lue.biosccc.uab.edu

Jianhua Huang
1061 Monroe St #7E
Albany, CA 94706
jianhua@stat.berkeley.edu

Sudha Jain
University of Toronto
Department of Statistics
Toronto, Ontario M5S 1A1 Canada
jainsu@utstat.utoronto.ca

Margaret K. James (Shaw)
Bowman Gray School of Med
Medical Center Blvd
Dept of Public Health Sciences
Winston-Salem, NC 27157
pjames@phs.bgsm.wfu.edu

Robert W. Jernigan
The American University
Department of Mathematics and Statistics
4400 Massachusetts Ave. NW
Washington, D.C. 20016-8050
jernigan@american.edu

Don Johns
Eli Lilly and Company
Lilly Corporate Center
Indianapolis, IN 46285

Mark A. Johnson
Upjohn Laboratories
301 Henrietta Street
Kalamazoo, MI 49007
majohns1@intnet.upj.edu

Gordon Johnston
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
sasgjj@unx.sas.edu

Elizabeth Johnston
Oxford University Press
Walton Street
Science Medical & Journals Division
Oxford OX2 6DP U. K.
oupeaj@ox.vax.ac.uk

Iain Johnstone
Stanford University
Department of Statistics
Sequoia Hall 94305-4065
imj@playfair.stanford.edu

Karen Kafadar
Univ of Colorado - Denver
P.O. Box 173364,
Box 170
Denver, CO 80217-3364
kk@helix.nih.gov

Al Kalantar
University of Alberta
Chemistry Department
Edmonton, Alberta T6G 2G2 Canada
kalantar@L.chem.ualberta.ca

Subramanyam Kasala
Univ N Carolina
Department of Mathematical Sciences
601 S College Rd
Wilmington, NC 28403

Charles Katholi
Univ of Alabama-Birmingham
Department of Biostatistics
School of Public Health, UAB Station
Birmingham, AL 35294-2030
katholi@cis.uab.edu

Sallie Keller-McNulty
Kansas State University
Statistics Department
Dickens Hall
Manhattan, Kansas 66506-0802
sallie@dutchess.stat.ksu.edu

Colleen Kelly
University of Rhode Island
Department of Computer Science and Statistics
Kingston, RI 02881
kelly@cs.uri.edu

David C. Kemp
Univ of St. Andrews, Scotland
Mathematical Institute
University of St Andrews N Haugh
St Andrews KY16 9SS Scotland
cdk@st-and.ac.uk

Freda (A.W.) Kemp
Univ of St. Andrews, Scotland
Mathematical Institute
University of St Andrews N Haugh
St Andrews KY16 9SS Scotland
awk@st-and.ac.uk

William Kemple
Naval Postgraduate School
OR/KE Room 239, Bldg. 302
Monterey, CA 93943-5000
kemple@NPS.NAVY.MIL

William J. Kennedy
Iowa State University
Statistical Lab
117 Snedecor
Ames, IA 50011
wjk@iastate.edu

Jon Kettenring
Bellcore
Bruno2C376
445 South Street
Morristown, NJ 07960-6438
jon@bellcore.edu

Barbara Keys
R.L. Polk, Inc
6400 Monroe
Taylor, MI 48180

Myrna M. Khan
Baylor College of Medicine
One Baylor Plaza
Houston, TX 77030-3915
mkhan@bcm.tmc.edu

Kyung Mann Kim
Harvard/Dana-Farber Cncr In
Harvard School of Public Health/Dana Farber
44 Binney Street, Mayer 4
Boston, MA 02115-6084
kkim@jimmy.harvard.edu

Alan P. Knoerr
Occidental College
Department of Mathematics
1600 Campus Road
Los Angeles, CA 90041
knoerr@oxy.edu

James Koehler
Univ of Colorado - Denver
P.O. Box 173364, C.B. 170
Denver, CO 80217-3364
jkoehler@copper.denver.colorado.edu

Robert Kohn
Univ of New South Whales
Australian Grad. School of Management
Kensington NSW 2033 Australia
robertk@mummy.agsm.unsw.oz.au

Eric D. Kolaczyk
Stanford University
Dept of Statistics
Sequoia Hall
Stanford, CA 94305
eric@playfair.stanford.edu

John E. Kolassa
University of Rochester
Box 630 Department of Biostatistics
University of Rochester Medical Center
Rochester, NY 14642
kolassa@metro.bst.rochester.edu

Martin Koschat
Yale University
School of Organization & Management
New Haven, CT 06520

Yuly Koshevnik
Southern Methodist University
Dallas, TX 75206

Andrzej S. Kozek
Macquarie University
Department of Statistics
Sydney, New South Wales 2109 Australia
akozek@zen.efs.mq.edu.au

Richard Krutchkoff
Virginia Polytech Inst & State U
Department of Statistics
Blacksburg, VA 24061
rgkrutch@vtucs.cc.vt.edu

Ann Kuo
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Soumendra Lahiri
Iowa State Univ
Deparment of Statistics
Ames, IA 50011

Alan Lapedes
Los Alamos National Labs
Complex Systems Group (T13)
Los Alamos, NM 87545
asl@t13.lanl.gov

Chip Lawrence
NCBI-NIH
Bethesada, MD
Lawrence@wadsworth.org

Patricia Lebow
US Forest Service
Forest Products Labs
One Gifford Pinchot Dr
Madison, WI 53705
patti@direwolf.fpl.wisc.edu

Tze-San Lee
Western Illinois University
Department of Mathematics
900 W. Adams Street
Macomb, IL 61455
MFTL@UXA.ECN.BGU.edu

Jaekyun Lee
University of Wisconsin - Madison
Department of Statistics
1210 W. Dayton St.
Madison, WI 53706
jaekyun@stat.wisc.edu

Dominic Lee
Johns Hopkins University
Mathematical Science Dept
Charles & 34th Streets
Baltimore, MD 21218
LDOMINIC@JHUVMS.HCF.JHU.edu

Bee-Leng Lee
National Univ of Singapore
Blk 563 ANG MO KIO Ave 3
#12-3443
Singapore-2056
ECSLEEBL%NUSVM.bitnet@CUNYVM.ed
u

Joseph Leighly
University of Washington
1111 E. Madison St., #256
Seattle, WA 98122
leighly@math.washington.edu

Raoul LePage
Univ of North Carolina (sab)
Department of Statistics
321 Phillips Hall
Chapel Hill, NC 27599

Wenlian Li
University of Waterloo
441 South First Street
#209
Ann Arbor, MI 48103
wli@stat.lsa.umich.edu

Charles Lin
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Mark Little
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
sasmrl@unx.sas.edu

Hong Liu
University of Pennsylvania
Department of Radiology
308 Stemmler Hall, 36th & Hamilton Walk
Philadelphia, PA 19104-6086
liu@optimizer.pndr.upenn.edu

Jun Liu
Harvard University
Department of Statistics, Science Center
10 DeWolf #46
Cambridge, MA 02138
jliu@stat.harvard.edu

Clive R. Loader
AT&T Bell Laboratories
600 Mountain Ave.
Room 2C-279
Murray Hill, NJ 07974
clive@research.att.edu

Michael D. Lock
Becton Dickinson
Immunocytometry Systems
2350 Qume Drive
San Jose, CA
Michael-Lock@bdis.edu

Paul J. Lucas
AT&T Bell Labs
1000 E. Warrenville Rd Room, 1P-336
P.O. Box 3013
Naperville, IL 60532
paul_j_lucas@att.edu

Cindy Lyles
NIEHS
PO Box 12233
RTP, NC 27709

Wayne Lytle
621 Theory Center
Cornell University
Ithaca, NY 14853
wayne@tc.cornell.edu

Steve MacEachern
Ohio State University
Statistics Department
148A Cockins Hall, 1958 Neil Ave.
Columbus, OH 43210-1247

David Marchette
NSWC
B10
Dahlgren, VA 22448
DMARCHE@RELAY.NSWC.NAVY.MIL

J.S. Marron
University of North Carolina
Department of Statistics
322 Phillips Hall, CB #3260
Chapel Hill, NC 27599-3260
marron@stat.unc.edu

Jonathan A. Marshall
University of North Carolina
Department of Computer Science
CB 3175
Chapel Hill, NC 27599
marshall@cs.unc.edu

Douglas R. Martin
MathSoft
1700 Westlake Ave., N
Suite 500
Seattle, WA 98109
doug@statsci.edu

Don Martin
University of Washington
SC-32, Biostatistics
Seattle, WA

Dan McCaffrey
Rand
1700 Main Street
Santa Monica, CA 90401

Melinda H. McCann
University of South Carolina
Department of Statistics
500 Harbison #2310
Columbia, SC 29212
mccann@stat.scarolina.edu

Mary M. McFarlane
University of North Carolina
110-A Misty Wood Circle
Chapel Hill, NC 27514
marymc@gibbs.oit.unc.edu

John D. McKenzie, Jr.
Babson College
Babson Park, MA 02157-0310
mckenzie@babson.edu

Nancy McMillan
NISS
P.O. Box14162
Research Triangle Park, NC 27709
mcmillan@niss.rti.org

Cyrus R. Mehta
Cytel Software Corporation
675 Massachusetts Ave
Cambridge, MA 02139
mehta@jimmy.harvard.edu

G. R. Mendieta
Wichita State University
Department of Mathematics
Wichita, KS 67260-0033

J. Alan Menius
Glaxo, Inc
Research Computing Department
5 Moore Drive P.O. Box 13358
Research Triangle Park, NC 27709
JAM22604@USAV01.GLAXO.edu

John R.Menkedick
Battelle
Statistics and Analysis System
505 King Ave
Columbus, OH 43212
menked@battelle.org

Peter Meyer
Rush Univ/Rush Presbyterian
St. Luke's Medical Center
1653 West Congress Parkway
Chicago, IL 60612-3824
meyer@bstat.pvm.rpslmc.edu

Michael Meyer
Carnegie Mellon University
HBH-3001-5000 Forbes Ave
Pittsburgh, PA 15213
mikem@stat.cmu.edu

Dan Meyer
Lubrizol Corporation
29400 Lakeland Blvd
Wickliffe, OH 44092
rdme@lubrizol.edu

James Mihalisin
Mihalisin Accociates Inc
600 Honey Run Road
Ambler, PA 19002

Roy Milton
National Eye Institute
11825 Gainsborough Road
Potomac, MD 20854
RCM@B31.NEI.NIH.GOV

Beverly Milton
Beverly
Guest-Roy Milton
11825 Gainsborough Road
Potomac, MD 20854

Xie Minge
Univ of Illinois
Champaign, Il

Salomon Minkin
Ontario Cancer Institute
Princess Margaret Hospital
500 Sherbourne Street
Toronto, Ontario M4X 1K9 Canada
Minkin@oci.utoronto.ca

Michael C. Minnotte
Utah State University
Department of Mathematics and Statistics
Logan, UT 84322-3900
minnotte@sunfs.math.usu.edu

Reza Modarres
George Washington Univesity
20133 Laurel Hill Way
Germantown, MD 20874
reza@GWUVM.GWU.edu

John Monahan
North Carolina State University
Statistics Department
Raleigh, NC 27695-8203
monahah@stat.ncsu.edu

Leslie M. Moore
Los Alamos National Lab
MS F600
707 4th Street
Los Alamos, NM 87544
lisa@emmy.lanl.gov

Wayne Moore
Stanford University Medical Center
Herzenberg Office, Dept of Genetics
Beckman Center, Room B-007
Stanford, CA 94305-5125

Richard Morris
Analytical Sciences Inc
Durham, NC 27713
morris@opie.niehs.nih.gov

Sally C. Morton
Rand
1700 Main Street
Santa Monica, CA 90406
Sally_Morton@rand.org

Bernard Most
ManTech Environmental
P.O.Box 12313
Research Triangle Park, NC 27709

Pierre Moulin
Bell Communications Research
Parallel Computing & Algorithms Group
445 South Street, Room 2M-393
Morristown, NJ 07960
moulin@bellcore.edu

Lawrence H. Muhlbaier
Duke University Medical Center
DUMC 3865
Durham, NC 27710-7510
muhl001@mc.duke.edu

Hans-Georg Muller
University of California-Davis
Statistics Department
469 Kerr Hall
Davis, CA 95616
hgmueller@ucdavis.bitnet

Peter Munson
National Institutes of Health
Analytical Biostatistics Section, DCRT
Lab of Structural Biology, Bldg 12A,Rm2041
Bethesda MD 20892
MUNSON@HELIX.NIH.GOV

Anupama Narayanan
SAS Institute Inc
SAS Campus Drive
Cary, NC
sasanu@unx.sas.edu

John Nash
University of Ottawa
Faculty of Administration
Ottawa, Ontario KIN 6N5 Canada
JCNASH@acadvm1.uottawa.ca

Ranjini Natarajan
Cornell University
230 Bryant Ave.
206 E&TC Building
Ithaca, NY 14850
ranjini@orie.cornell.edu

David Nelson
Lawrence Livermore National Lab
Biology and Biotechnology Research Program
Box 808, L-452
Livermore, CA 94550
daven@stille.llnl.gov

Gordon E. Nelson
AT&T Bell Labs
600 Mountain Ave, Room 3C-513
P.O. Box 636
Murray Hill, NJ 07974-0636

Padraic Neville
Data Sprouts
P.O. Box 114
Port Costa, CA 94569-0114

David Newman
Boeing Computer Services
P.O. Box 24346, MS 7L-22
Seattle, WA 98124-0346
dnewman@espresso.rt.cs.boeing.edu

Joseph H. Newton
Texas A&M University
Department of Statistics
447 Blocker Building
College Station, TX 77843
jnewton@stat.tamu.edu

Katherine Ng
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Sharon-Lise T. Normand
Harvard Medical School
Department of Health Care Policy
Parcel B-1st Floor, 25 Shattuck St
Boston, MA 02115
sharon@Figaro.med.harvard.edu

Douglas W. Nychka.
North Carolina State University
P.O. Box 8203
Raleigh, NC 27695-8203
nychka@stat.ncsu.edu

Michael O'Connell
Becton Dickinson
P.O. Box 12016
Research Triangle Park, NC 27709
moc@bdrc.bd.edu

Art B. Owen
Stanford University
Statistics Department
Sequoia Hall
Stanford, CA 94305
owen@playfair.stanford.edu

Panickos Palettas
Virginia Tech
Department of Statistics
Blacksburg, VA 24061-0439

Emanuel Parzen
Texas A&M University
Statistics Department
College Station, TX 77843-3143
eparzen@stat.tamu.edu

Ranjit M. Passi
COAM/USM SSC, MS
Stennis Space Center, MS 39529
passi@coam.usm.edu

Miroslaw Pawlak
University of Manitoba
Departmetn of Electr & Comp. Eng.
Winnipeg, Manitoba R3T 5V6 Canada
Pawlak@eeserv.ee.umanitoba.ca

Mario Peruggia
Ohio State University
129B Cockins Hall
1958 Neil Avenue
Columbus , Ohio 43210-1247
peruggia@mps.ohio.state.edu

Lori Pfahler
Rohm & Haas
727 Norristown Rd.
Spring House, PA 19477
rs0lbp@rohmhaas.edu

Philippe Castagliola
Ecole des Mines de Nantes
3 rue Marcel Sembat
44040 NANTES
Cedex 04 France
pcasta@auto.emn.Fr

Jane Pierce
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Bill V. Pikounis
MERCK Research Labs
P.O. Box 2000 RY70-38
Rahway, NJ 07065
V_bill_p@merck.edu

Joerg Polzehl
Konrad-Zuse-Zentrum f Inform.
Heilbronner Strasse10
Berlin, Wilmersdorf D-10711 Germany
polzehl@zib-berlin.de

Joerg Polzehl
Konrad-Zuse-Zentrum f Inform.
Heilbronner Strasse10
Berlin, Wilmersdorf D-10711 Germany
polzehl@zib-berlin.de

Chris Portier
NIEHS
P.O. Box 12233
Research Triangle Park, NC 27709
portier@milo.niehs.nih.gov

Wendy Poston
Naval Surface Warfare Center
NSWCDD G33, B10
Systems Research/Tech Department
Dahlgren, VA 22448
wposton@pooh.nswc.navy.mil

Daryl Pregibon
AT&T Bell Labs
Room 2C-264
Murray Hill, NJ 07974
DARYL@RESEARCH.ATT.edu

Carey Priebe
Naval Surface Warfare Center
Systems Research/Tech Dept
Advanced Computation Tech, B10
Dahlgren, VA 22448
cpriebe@relay.nswc.navy.mil

Shixian Qian
Carnegie Mellon University
17 Welsford Street
Pittsburgh, PA 15213
qian@stat.cmu.edu

Song Qian
Duke Univ Schl Environment
Box 90382
Durham, NC 27708-0328
song@isds.duke.edu

Adrian Raftery
University of Washington
Statistics Department, GN-22
Seattle, WA 98195
RAFTERY@STAT.WASHINGTON.edu

Robert Read
Naval Postgraduate School
Code OR/Re
Monterey, CA 93943

David Reboussin
Bowman Gray Schl of Medicine
Medical Center Blvd
Winston-Salem, NC 27157
davidr@hugh.bgsm.wfu.edu

Kenneth H. Reckhow
Duke University
Box 90328
Durham, NC 27708-0328
reckhow@acpub.duke.edu

Rich Richardson
Univ of Texas-San Antonio
Department of Mathematics
San Antonio, TX 78249
rich@ssdt-bluestein.sps.mot.edu

Christian Ritter
Univ Catholique de Louvain
Rue Mercelis 5
B-1050 Brussels
Belgium
ritter@stat.ucl.ac.be

Gareth O. Roberts
University of Cambridge
Statistical Laboratory
16 Mill Lane
Cambridge CB2 1SB U.K.
G.O.Roberts@statslab.cam.ak.uk

Heman Robinson
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

David Rocke
University of California-Davis
Graduate School of Management
Davis, CA 95616-8609
dmrocke@ucdavis.edu

Robert Rodriguez
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
sasrnr@unx.sas.edu

Charles Roosen
Stanford University
Department of Statistics
Sequoia Hall
Stanford, CA
94305

James L Rosenberger
Penn State University
Statistics Department
214 Pond Lab
University Park, PA 16802-2111
jlr@stat.psu.edu

Jeffrey Rosenthal
University of Toronto
Department of Statistics
Toronto, Ontario M5S 1A1 Canada
jeff@utstat.toronto.edu

Gary L Rosner
Duke University Medical Cntr
Box 3958
Durham, NC 27710
grosner@ccstat.mc.duke.edu

Matt Rotelli
Virginia Tech
750 Hunter Mill Rd
Apt 9200 G
Blacksburg, VA 24060
ROTEL@VTVMI.CC.VT.edu

Matt Rotelli
Virginia Tech
750 Hunter Mill Road
Apt 9200G
Blacksburg, VA 24060
rotel@vtvm1.cc.vt.edu

Chan Russell
Belmont Research
84 Sherman Street
Cambridge, MA 02140

Bert W. Rust
Natn'l Inst Standards &Tech
Applied and Computational Mathematics Div
Bldg 101 Room A238
Gaithersburg, MD 20899
bwr@cam.nist.gov

Shiva K Saksena
Univ of North Carolina-Wilmington
Department of Math Sciences
601 S. College Road
Wilmington, NC 28403

John Sall
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513
sall@sas.edu

Warren Sarle
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Michael G. Schimek
Univ of Graz Medical Schools
Medical Biometrics Group
Auenbruggerplatz 30/IV
Graz A-8036 Austria
schimek@bkfug.kfunigraz.ac.at

David Schlotzhauer
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Christopher H. Schmid
Tufts Univ/New England Med Cntr
Box 63, 750 Washington Street
Boston, MA 02111
cschmid@opal.tufts.edu

David W. Scott
Rice University
12123 Gray Star Way
Columbia, MD 21044
scottdw@rice.edu

David J. Scott
Colorado State University
Department of Statistics
Fort Collins, CO 80525
scott@tlaloc.stat.colostate.edu

Mark Segal
Univ of California-San Francisco
Biostatistics Department
Box 0560
San Francisco, CA 94143-0560
mark@biostat.ucsf.edu

Burkhardt Seifert
University of Zurich
Department of Biostatistics ISPM
Sumatrastrasse 30
CH-8006 Zurich
Switzerland
seifert@ifspm.unizh.ch

Francoise Seillier-Moiseiwitsch
University of North Carolina
Department of Biostatistics
CB 7400
Chapel Hill, NC 27599-7400
seillier@biostat.sph.unc.edu

David N. Sessions
9 Purdue Road
Glen Cove, NY 11542

Nong Shang
University of California, Berkeley
Dept Biomedical & Environmental Health Sci
3400 Richmond Pkwy #418
Richmond, CA 94806
shang@stat.berkeley.edu

Yuehjen Eric Shao
Fu Jen University
Dept of Statistics
College of Management
Taipei
Taiwan, R.O.C.
sta104@fju.edu.tw

Yuehjen Eric Shao
Fu Jen University
Dept of Statistics
College of Management
Taipei
Taiwan, R.O.C.
sta104@fju.edu.tw

Frank Shen
MERCK Company
P.O. Box 2000
RY70-38
Rahway, NJ 07065

Claire Sherman
NIEHS
Po box 12233
Research Triangle Park, NC 27709

Chen-chi Shing
Radford University
Computer Science Department
Box 6933
Radford, VA 24142
cshing@rucs.faculty.cs.runet.edu

Douglas Simpson
Univ of Illinois
Dept of Statistics101 Illini Hall
725 S. Wright St
Champaign, IL 61820

Carolyn Sistar-Magri
Dept of Defense DOD
Ft. Meade, MD 20755
carolyn@zombie.ncsc.mil

Elizabeth Slate
Cornell University
Operations Research & Industrial Engineering
228 ETC Building
Ithaca, NY 14853
slate@orie.cornell.edu

Smith Richard
University of North Carolina
Statistics Department
CB#3260 322 Phillips Hall
Chapel Hill , NC 27599-3260
rs@stat.unc.edu

Adrian F.M. Smith
Imperial College of London
Department of Mathematics
Imperial College
London SW7 2BZ England
a.smith@uk.ac.ic

William B. Smith
Texas A&M University
Statistics Department
College Station, TX 77843
smith@pkard.tamu.edu

Patricia Smith
Guest-William Smith
1040 Rose Circle
College Station, TX 77840

Shannon L Smosarski
Sunny New Platz
Stone Ridge, NY 12484

Gordon K. Smyth
University of Queensland
Department of Mathematics
St. Lucia Q 4072 Australia
gks@maths.uq.oz.au

Ying So
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Jeff Solka
Naval Surface Warfare Center
Systems Research and Technology Dept
Advanced Computation Technology Group
Code B10
Dahlgren, VA 22448-5000

Paul Somerville
University of Central Florida
Department of Statistics
Orlando, FL 32765
somer@cs.ucf.edu

Matthew C. Somerville
ManTech Environmental Tech Inc
2 Triangle Drive
Research Triangle Park, NC 27709
mza@nccibm1.bitnet

Paul Speckman
University of Missouri-Columbia
Statistics Department
318 Math Sci Bldg
Columbia, MO 65211
statpls@umcvmb

Phil Spector
University of California
Statistics Department
Berkeley, CA
spector@gandalf.Berkeley.edu

Terence Speed
Univ of California, Berkeley
Statistics Department
Berkeley, CA 94720
terry@stat.Berkeley.edu

Dalene K. Stangl
ISDS, Duke
Durham, NC 27708-0251
dalene@isds.duke.edu

Andrew G. Stead
Organon Teknika Corporation
100 Akzo Avenue
Durham, NC

Dan Steinberg
Salford Systems
5952 Bernadette Lane
San Diego, CA 92120

Laura Steinberg
Ntnl Inst Statistical Sciences
P.O. Box 14162
Research Triangle Park, NC 27709-4162
ljs@niss.rti.org

G.W. Stewart
University of Maryland
Computer Science Department
College Park, MD 20742
stewart@cs.umd.edu

Richard J.Stewart
UNC Hwy Safety Rsrch Cntr
134 1/2 E. Franklin
Chapel Hill, NC 27599

Tom Stockton
Duke University
16843 W. 75th Place
Golden, CO 80403
barber@acpub.duke.edu

Maura Stokes
SAS institute Inc
SAS Campus Drive
Cary, NC 27513
sasmzs@unx.sas.edu

Quentin Stout
University of Michigan
EECS Department
Ann Arbor, MI 48109-2122
qstout@eecs.umich.edu

Arnold Stromberg
University of Kentucky
Department of Statistics
817 Patterson Office Tower
Lexington, KY 40506-0027
astro11@ukcc.uky.edu

Walter B. Studdiford
Princeton University (retired)
170 Old York Rd
Bridgewater, NJ 08807-2629
C1379@PUCC.PRINCETON.edu

Deborah Sturm
The College of Staten Island
Computer Science Department 1N 207
2800 Victory Boulevard
Staten Island, NY 10314
ddssi@cunyvm.cuny.edu

Clifton D. Sutton
George Mason University
Center for Computational Statistics
Fairfax, VA 22030
sutton@mason1.gmu.edu

Deborah Swayne
Bellcore
Statistics and Data Analysis Research Group
445 South Street , Room 2L331
Morristown, NJ 07960
dfs@bellcore.edu

Juergen Symanzik
Iowa State University
Department of Statistics
Ames, Iowa 50011
symanzik@iastate.edu

Terry V. Taerum
University of Alberta
Edmunton, Alberta T6G 2E1 Canada
ttaerum@mts.ucs.ualberta.ca

Nader Tajvidi
Chalmers Univ of Technology
Department of Mathematics, CTH
Gothenburg 412 96
Sweden
nader@math.chalmers.se

Ming Tan
Cleveland Clinic Foundation
Department of Biostatistics Desk P88
9500 Euclid Ave
Cleveland, OH 44195
mtan@bio.ri.ccf.org

Martin Tanner
Martin
Univ of Rochester Medical Center
Department of Biostatistics
Box 630
Rochester, NY 14642
tanner@uorhbv.bitnet

Mike Tarter
University of California
Biomed Env Hs Dept
32 Earl Warren Hall
Berkeley, CA 94720
tarter@gandalf.Berkeley.edu

Robert F. Teitel
Abt Associates Inc
4800 Montgomery Lane
Bethesda, MD 20814

George Terrell
Virginia Polytech Inst &State U
Statistics Department
Blacksburg, VA 24061-0439
terrell@vtvm1.cc.vt.edu

Peter F. Thall
M.D. Anderson Cancer Center
Department of Biomathematics
Box 237, 1515 Holcombe Blvd
Houston, TX 77030
lunch@odin.mda.uth.tmc.edu

Terry Therneau
Mayo Foundation
Health Science Research Department
200 First Street SW
Rochester, MN 55905
therneau@mayo.edu

Andrew Thomas
Inst of Public Health, Cambridge
MRC Biostatistics Unit
Robinson Way
Cambridge, CB2 2SR U.K.
andrew.thomas@mrc-bsu.aom.ac.uk

Elizabeth Thompson
University of Washington
Statistics Department, GN-22
Seattle, WA 98195
thompson@stat.washington.edu

Rob Tibshirani
University of Toronto
Prev Med & Biostat Dept
Toronto M5S 1A8 Canada

Dave Tilley
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

John Timar
Polysar Rubber
P.O. Box 3001
1265 Vidal Street South
Sarnia, Ontario N7T 7M2 Canada

Dimitri Tischenko
Delft University of Technology
Dept of Applied Math & Computer Science
Mekelweg 4
Delft
2628 CD
The Netherlands
D.B.Tischenko@TWI.TUDelft.NL
Tischenko

Rodney A. Tjoelker
Boeing Computer Services
P.O. Box 24346 MS 7l-22
Seattle, WA 98124-0346
tjoelker@espresso.rt.cs.boeing.edu

Randall Tobias
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

David Tritchler
Ontario Cancer Institute
500 Sherbourne Street
Toronto, Ontario M4X 1K9 Canada
tritchle@oci.utoronto.ca

Berwin A.Turlach
Univ Catholique de Louvain
CORE and Institute de Statistique
34, Voie du Roman Pays
1348 Louvain-la-Neuve
Belgium
turlach@core.ucl.ac.be

Pauline Vaas
Duke University
Durham, NC

L.D. Van Vleck
University of Nebraska
A-218 Animal Sciences complex
Lincoln, NE 68583-0908
ANSC418@UNL.VM

Dominic F. Vecchia
Natn'l Inst Standards & Tech NIST
325 S. Broadway
Boulder, CO 80303
dfve@bldrdoc.gov

John E. Vetter
Department of Navy
200 N. Pickett Street
Alexandria, VA 22304
[vetterje@am@nimitz]:3cgate:navair

Brani Vidakovic
Duke University
ISDS
Box 90251
Durham, NC 27708

Edward F. Vonesh
Baxter HealthCare Corporation
Applications Statistics Center
Route 120 & Wilson Road
Round Lake, IL 60073-0490

Carrie Wager
Channing Laboratory
180 Longwood Ave
Boston, MA 02115
recgw@gauss.med.harvard.edu

Morgan C. Wang
University of Central Florida
Department of Statistics
UCF
Orlando, FL 32828
FDCWANG@UCFIVM

Suojin Wang
Texas A&M University
Department of Statistics
College Station, TX 7843-3143
sjwang@stat.tamu.edu

Ruey Wang
Olin Corp
350 Knotter Drive
Cheshire, CT 06410

Yazhen Wang
University of Missouri-Columbia
Department of Statistics
Columbia, MO 65211
yzwang@stat.missouri.edu

Jiannong Wang
Memphis State University
Department of Mathematical Sciences
Memphis, TN 38111
wangj@hermes.msci.memst.edu

Morgan C. Wang
University of Central Florida
Department of Statistics
Orlando, FL 32828
FDCWANG@UCFIVM

Minhwei Wang
Educational Testing Service
Princeton, NJ 08541-0001
mhwang@rosedale.org

David G. Ward.
Barron Associates, Inc
3046A Berkmar Dr
Charlottesville, VA 22901-1444
bai@attmail.edu

Christine Waternaux
Harvard University
Biostat Department, School of Public Health
677 Huntington Ave.
Boston, MA 02115

Edward Wegman
George Mason University
Center for Computational Statistics
157 Science-Technology Bldg #2
4400 University Drive
Fairfax, VA 22030
ewegman@endor.galaxy.gmu.edu

William J. Welch
University of Waterloo
Statistics and Actuarial Science
Waterloo N2L 3G1 Canada
wjwelch@watstat.uwaterloo.ca

Mike West
Duke University
Institute of Statistics and Decision Sciences
Box 90251
Durham, NC 27706
mw@isds.duke.edu

P.H. Westfall
Texas Tech University
Info Systems and Quant Sci
Mail Stop 2101
Lubbock, TX 79409
Bitnet%"ODWES@TTACS1"

Christopher Wiesen
University of North Carolina
Department of Statistics
Chapel Hill, NC 27599

Valerie S.L. Williams
Ntn'l Inst Statistical Sciences
P.O. Box 14162
Research Triangle Park, NC 27709
williams@niss.rti.org

Calvin L. Williams
Clemson University
Box 341907
Dept of Math Sciences
Clemson, NC 29634-1907
calvin@clust1.clemson.edu

David G. Williamson
Centers for Disease Control
EPO/DSE/SAMB, MS-G34
1600 Clifton Rd NE
Atlanta, GA 30333
CDCGDW@EMUVM1.Bitnet

Graham J. Wills
AT&T Bell Labs
Room 1U-334
1000 E. Warrenville Road, PO Box 3013
Naperville, IL 60566
gwills@research.att.edu

Russell Wolfinger
SAS Institute Inc
SAS Campus Drive
Cary, NC 27513

Peter Wollan
Mayo Clinic
200 First Street Southwest
Rochester, MN 55905
wollan@mayo.edu

Terry J. Woodfield
Risk Data Corporation
111 Pacifica 3rd Floor
Irvine, CA 92718-3331

David Woodruff
Research Department, ACT
P.O. Box 168
2201 N Dodge St
Iowa City, IA 52243

Trong Wu
S. Ilinois Univ-Edwardsville
Department of Computer Science
Bldg II, Room 230
Edwardsville, IL 62026-1656
twu@siuemus.bitnet

Momiao Xiong
University of Southern California
Department of Mathematics and Molecular Biology
Los Angeles, CA 90089

Lu Xu
Proctor & Gamble Pharm
Regulatory and Clinical Development
Sharon Woods Tech Center
Cincinnati, Ohio 45241-2422
procter!xu-1!ms.uky.edu

Yaqi Yang
Univ of N Carolina-Wilmington
Mathematics Department
Wilmington, NC 28403
yang@seq.cms.uncwil.edu

Kwang-Su Yang
George Mason University
Center for Computational Statistics
157 Science-Technology Building #2
Fairfax, VA 22030

Stanley S. Young
Glaxo, Inc
Research Computing Department
5 Moore Drive P.O. Box 13358
Research Triangle Park, NC 27709
SSY0487@usav01.glaxo.edu

Alastair G. Young
University of Cambridge
Statistics Lab
16 Mill Lane
Cambridge CB2 1SB U.K.
G.A.Young@statslab.cam.ac.uk

Forrest Young
University of North Carolina
UNC Psychometrics
CB-3270 Davie Hall
Chapel Hill, NC 27599-3270
uluru@unc.edu

Heping Zhang
Yale University School of Medicine
Department of Epidemiology and Public Health
New Haven, CT 06510
heping@peace.med.yale.edu

Haibo Zhou
National Inst Statistical Sciences
P.O. Box 14162
Research Triangle Park, NC 27709-4162
haibo@biostat.sph.unc.edu

SYMPOSIUM SESSION SCHEDULE

Thursday June 16, 1994

8:15 a.m. - 9:45 a.m.
Keynote Session

- Gauss, Statistics, and Gaussian Elimination

10:15 a.m. - 12:00 p.m.
Issues in Software

- Software as Property
- Developing Interactive Graphics in C++
- Parallel Computing And Statistics

10:15 a.m. - 12:00 p.m.
Fast Implementations of Smoothers

- Fast Implementations of Nonparametric Curve
- Estimation and Presentation of Regression in Several Variables via Warping and the ASH
- Fast and Stable Computation of Local Polynomials
- Fast Implementations of Average Derivative Estimation

10:15 a.m. - 12:00 p.m.
Longitudinal and Mixed Models,

- Estimation Methods for Nonlinear Mixed-Effects Models
- Experiences with Derivative-Free REML for Large, Messy, Multiple Trait Genetic Models to Estimate Variances and Covariances
- Generalized Estimating Equations and Extensions for Various Clustered Data Structures

10:15 a.m. - 12:00 p.m.
Contributed Papers 1: Experimental Design

- Dual Space Algorithms for Designing Space Filling Experiments
- Experiment Design for Assessment of Important Inputs to a Computer Code
- Computations in a Finite Projective Geometry for Enumeration of Subdesigns
- Sampling Plans on the Sphere

10:15 a.m. - 12:00 p.m.
Contributed Papers 2: Fractal, Neural, other

- Incorporating Segmentation Boundaries into the Calculation of Fractal Dimension Features
- Overfitting in Neural Networks
- Likelihood Profiles for Studying Non-Identifiability
- A Method for Estimation of Parameters of the Keeney & Raifa Utility Models Based on the Normal Logistic Functions
- Statistical Fitting of Financial Models

12:45 p.m. - 1:30 p.m.
POSTER SESSIONS

- On Calculating the Distribution of Independent Trials with Changing Probabilities of Success
- Bayesian Estimation Using the Gibbs Sampler for the Inhibition/Promotion Cancer Chemoprevention Experiment
- Computationally Intensive Statistical Methods for Quality Control
- Interval Analysis and Self-Validating Computation of Non-Central F Probabilities and Percentiles,
- An Algorithm for Fitting and Displaying Distribution Data
- MCMC Methods When There Is Partial Exchangeability
- Graphically Comparing Two Similarity Measures Defined over Large Databases
- Robust Empirical and Hierarchical Bayes Estimation of Normal Means and Rates in Longitudinal Studies

1:30 p.m. - 3:15 p.m.
Green Thumbs: Extensions and Applications of Tree Modeling Methods,

- The Art of Growing Classification Trees
- Trees for Event Rate Data
- Hybrid Trees

1:30 p.m. - 3:15 p.m.
Bayesian Curve Fitting

- Gibbs Sampling schemes for Bayesian Density Estimation with Mixtures
- Nonparametric Additive Regression with Autocorrelated Errors
- Issues in Bayesian Analysis of Neural Networks
- Wavelets and Bayesian Data Analysis

1:30 p.m. - 3:15 p.m.
Space Filling Experimental Designs: Theory, Computer Construction, and Analysis

- Introduction to Space Filling Designs
- Algorithms and Uses of Space Filling Designs
- Analysis of Space Filling Designs

1:30 p.m. - 3:15 p.m.
Contributed Papers 3: Longitudinal

- A Monte Carlo E M Algorithm for Some Grouped and Partially Observed Data Models with Random Effects: Ordinal Probit, Censored Regression and Tobit Models
- A Randomization Test for Diverging Trends in Longitudinal Data
- Linearizing Transformations in Growth-curve Problems
- An EM Algorithm Fitting First-Order Conditional Autoregressive Models to Longitudinal Data
- REML in Generalized Linear Models: A Conditional Approach

1:30 p.m. - 3:15 p.m.
Contributed Papers 4: Computing

- Random Integration Rules for Statistical Computation
- Using PVM on Computation for Analysis of Repeated Measurement Designs
- Large Visualizing Time-Stamped Log Files
- The Multi-String Rearranging Memory and Its Use in Statistical Computing
- Fast Multidimensional Density Estimation based on Random-width Bins

3:45 p.m. - 5:30 p.m.
Panel: Statistics Education in the Computer Age

3:45 p.m. - 5:30 p.m.
Contributed Papers 5: Enhancements to Tree Algorithms

- Multivariate Split Classification Trees
- Global Tree Optimization: a non-greedy decision tree algorithm
- Growing Decision Trees less Greedily
- Tree Structured Density Estimation
- Tree-Structured Multivariate Density Estimation and Its Application In Environmental Modeling

3:45 p.m. - 5:30 p.m.
Contributed Papers 6: Multiple Comparisons

- SIMMCOMP: an Splus Module for Simultaneous Inference
- Classical Multiple Comparison via Naiman's Inequality From Hypercubes to Permutation Polytopes
- On The Analysis of Multiple Correlated Binary Endpoints in Medical Studies

3:45 p.m. - 5:30 p.m.
Contributed Papers 7: Smoothers and Nonparametric Regression

- On Partial Cross Validation in Nonparametric Regress
- An Iterative Projection Method for Nonparametric Additive Regression Modeling
- Nonparametric Curve Estimation from Indirect Observations
- Open Questions in the Application of Smoothing Methods to Finite Population Inference
- Empirical Examination of an Efficient Robust Linear Regressor

3:45 p.m. - 5:30 p.m.
Tutorial: Introduction to Perl, for Statisticians

Friday June 17, 1994

8:15 a.m. - 9:45 a.m.
Neural Nets

- The Accuracy of Bayes Estimators of Neural Nets
- Using neural networks to estimate functions
- Generalization and Exclusive Allocation in Unsupervised Category Learning

8:15 a.m. - 9:45 a.m.
Contributed Papers 8: Density Estimation

- Jump and Sharp Detection by Wavelets
- Numerical Techniques in Distribution Fitting
- Maximum Likelihood Density Estimation with Term Creation and Annihilation
- The Bias-Optimized Frequency Polygon
- Data Adaptive Density Estimation of DNA Distributions

8:15 a.m. - 9:45 a.m.
Contributed Papers 9: Numerical

- Parseval Quadrature for Normal Tail Possibilities
- A Method of the Computation of Multivariate Normal Probabilities Over Any Convex Region
- Computer Random Variate Generation for Multinomial Distribution
- Systematic Random Leapfrog Method for Parallel Random Number Generators
- Efficient Programs for Simulating Chi-bar Square Distributions

8:15 a.m. - 9:45 a.m.
Contributed Papers 10: Trees II

- The Cumulative Score Control Chart for an Open Loop Control
- Piecewise Proportional Hazards Survival Trees With Time -Dependent Covariates
- A Tree-Based Method of Analysis for Prospective Studies

vTesting in High Dimensional Spaces via Recursive Partitioning

- Tree Based Classification Using a Predictor with Many Categories

8:15 a.m. - 9:45 a.m.
Tutorial: Networking Innovations and Resources, The Internet as Toolbox

10:15 a.m. - 12:00 p.m.
Nonparametric Regression for Edge and Peak Preserving

- Cube splitting for multidimensional edges
- Discontinuity estimation in nonparametric regression via orthogonal series
- Semiparametric Change-Point Methods
- Nonparametric autoregression-regression for edge preserving: The estimate and its application in computer vision

10:15 a.m. - 12:00 p.m.
Software for MetaAnalysis

- Epi-meta: Meta-analytic statistical software for epidemiological studies
- Performing meta-analyses using commercial mixed-model software
- Software for meta-analysis: a comparative review

10:15 p.m. - 12:00 p.m.
Special Contributed Papers 11: Visual Statistical Analysis

- Visually Guided Statistical Analysis
- Visual Sensitivity Analysis for Multidimensional Scaling
- Visual Correspondence Analysis
- Visual Log-Linear Analysis
- Visualizing High-Dimensional Space with Principal Components Analysis

10:15 p.m. - 12:00 p.m.
Contributed Papers 12: Gibbs Samplers

- Monte Carlo Assessment of Influence and Sensitivity in Bayesian Modeling
- Bayesian Inference for Nonlinear Regression with Covariate Measurement Error via Gibbs Sampling
- BUGS (Bayesian Inference Using Gibbs Sampling)
- Using the Gibbs Sampler to Detect Changepoints: Application to Longitudinal Markers of Disease
- Applied Convergence Diagnostics for the Gibbs Sampler

10:15 p.m. - 12:00 p.m.
Contributed Papers 13: Computing

- Statistical Inference for Priority Queues
- On-Line Control of Stochastic Systems: Application to the Design of an Artificial Pancreas
- Using both Symbolic and Classical Methods to Analyze Complex Data Set with the SAS System
- Statistical Methods in Software Engineering

1:30 p.m. - 3:15 p.m.
Wavelets Tutorial

1:30 p.m. - 3:15 p.m.
Smart Monte Carlo Methods for Conditional Inference in Exponential Families

- Approximate conditional inference in exponential families via the Gibbs sampler
- Saddlepoint approximations for the likelihood ratio statistic in exponential families
- Monte Carlo sampling from exponential families under linear constraints

1:30 p.m. - 3:15 p.m.
Stochastic Modeling In Carcinogenesis

- Computational issues in analyzing premalignant liver lesions
- Multi-pathway multistage models of carcinogenesis
- Time-dependent rates in interconnected birth-death models

1:30 p.m. - 3:15 p.m.
Contributed Papers 14: Multivariate

- Stability of Homogeneity Analysis
- Estimation of Covariance Matrices Using Eigenstructure Influence
- Finding the Minimum Volume Ellipsoid
- Triangulation and Multivariate Nonparametric Function Estimation

1:30 p.m. - 3:15 p.m.
Contributed Papers 15: Software

- Data Conversion Pitfalls
- Design of Object-Oriented Functions in S for Screen Display, Interface and Control, of Other Programs
- LISP for Interval Computations
- Documentation with Online Programs Rather Than Programs with Online Documentation
- What is the Most Appropriate Software for a Statistics Course?, John D. McKenzie, Jr., and William H. Rybolt, Babson College

3:45 p.m. - 5:30 p.m.
Panel of Editors of Journals for Statistical Computing

3:45 p.m. - 5:30 p.m.
Robust Regression and Multivariate Analysis

- Using Multiple Processors to Compute Robust Regression Estimators
- Identification of Outliers in Multivariate Data
- Robust Model Comparison for Autoregressive Processes with Robust Bayes Factors

3:45 p.m. - 5:30 p.m.
Applications of Wavelets

- S+WAVELETS: An Object-Oriented Wavelet Toolkit
- The Use of Wavelets for Spectral Density Estimation With Local Bandwidth Adaptation
- An Application of Wavelets to Tomography
- Use of Wavelets for Denoising and Feature Enhancement in Mammograms

3:45 p.m. - 5:30 p.m.
Contributed Papers 16: Genetics

- A Composite Model for the Distribution of Species and Its Use in Monitoring Pattern Recognition Algorithms
- Some Computational Problems in Modeling Molecular Evolution
- Computation of Identity-by-Descent Proportion for Pedigree Data
- Using S for a Bayesian Analysis of Cleavage Sites When the Amino Sequence in a Peptide Is Known
- Inference for Lethal Gene Studies via Bayesian Markov Chain Simulation

3:45 p.m. - 5:30 p.m.
Contributed Papers 17: Bootstrap

- Aggregation Coefficients of Clustering in Databases and Metric Spaces
- On a Nearest Neighbours Oriented Algorithm for Missing Data Reconstitution-Application to a Magamatic Data Array
- Efficient Computation of Statistical Procedures Based on Subsetting the Observations
- A Frequency Domain Bootstrap for Time Series

Saturday June 18, 1994

8:15 a.m. - 9:45 a.m.
Tutorial: Markov Chain Monte Carlo in Bayesian and Likelihood Statistics

8:15 a.m. - 9:45
Statistics of Protein and Macromolecular Structures

- Threading protein sequences through folding motifs
- The Inverse Folding Problem: Analysis by Statistical and Machine Learning Methods
- A Gibbs sampling algorithm for the identification and characterization of a structural motif in a data base of bioploymer sequences.

8:15 a.m. - 9:45 a.m.
Contributed Papers 18: Graphics

- Visualizing the Destructive Potential of Indirect Fire Weapons
- A Robust Visual Access and Analysis System for Very Large Multivariate Databases
- Dynamic Graphics in a GIS: A Link between Arc/Info and XGobi
- Variations on Row-labeled Plots
- Data Analysis with Graphical Models

8:15 a.m. - 9:45 a.m.
Contributed Papers 19: Nonparametrics

- Relative Power of Smirnov and Wilcoxon exact tests in two-sample .
- A Simulation Study of Some Rank Tests for Interaction in Two-Way Layouts
- Computation of the Wilcoxon's T(n), Wilcoxon's W(m,n) and Ansari-Bradley's A(m,n) Statistics When the Sample Size is Small
- NonParametric Estimation of Functions from Stratified Samples

10:15 a.m. - 12:00 p.m.
Efficient Bootstrap Computations

- Concomitants of order statistics for bootstrap distribution estimation
- Fast and accurate approximate double bootstrap confidence intervals
- Saddlepoint Control Variates and Importance Sampling

10:15 a.m. - 12:00 p.m.
Convergence of Markov Chain Samplers

- Efficient Random-Walk Metropolis Algorithms
- Theoretical rates of convergence for Markov chain Monte Carlo
- The fraction of missing information and convergence rate for data augmentation

10:15 a.m. - 12:00 p.m.
Computational Techniques in Genetics and Molecular Biology

- Monte Carlo Estimation of Autozygosity Probabilities, Elizabeth Thompson, Univ of Washington;
- Statistical and Computational Challenges in Physical Mapping, David Nelson & Terence Speed, Berkeley;
- Bayesian Restoration of a Hidden Markov Chain with Application to Sequence Alignment,Gary Churchill, Cornell.

10:15 a.m. - 12:00 p.m.
Contributed Papers 20: Robust

- Regression Hazards Model with Markov Process
- A Principal Components Based Algorithm for Variable Selection in Linear Models
- Saddlepoint Approximations for Robust M Regression
- Rank Cusum Test for Change in the Mean

10:15 a.m. - 12:00 p.m.
Contributed Papers 21: Parametric Modeling

- Perturbation Bounds for Linear Regression Problems
- Tailoring Nonlinear Least Squares Algorithms for the Analysis of Compartment Models
- Characterizing Hierarchical Model Behavior
- Predicting the Urban Ozone Levels and Trends with Semiparametric Modeling